DIPLOMA-THESIS

nCEA - natural Context Exploration Approach

A proposal for a pragmatic approach to modeling and managing business processes and corresponding IT-projects

by

Matthes Elstermann

Submitted on the 26th of August 2010 to the
Institute of Applied Informatics and

Formal Description Methods (AIFB)

at

Referee: Prof. Dr. Detlef Seese

Supervising Tutor: Dipl. Wi.-Ing. Hagen Buchwald

Address:

Matthes Elstermann

Friedrich-Karl-Str. 1

68165 Mannheim

e-Mail: matthes.elstermann@gmail.com

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Diplomarbeit selbständig und ohne unerlaubte Hilfsmittel angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

[I hereby declare, that this diploma thesis was written independently and without the use of illegal help, that nothing but the listed source and tools were used, and that sections taken literally or by content from those sources are indicated as such. The thesis, neither in equal nor similar form, nor in excerpts, has not yet been submitted to another board of examination, nor has it been published.]

Karlsruhe - 26. August 2010

Matthes Elstermann

Abstract

This thesis is concerned with the question how existing running and robust business processes can be mapped, modeled, measured, and via those actions how they can be conveyed into a continuous self-improvement process-framework.

A process-approach solution will be developed that is called nCEA – the natural Context Exploration Approach.

It will be derived from several ideas that are reviewed and discussed. This includes existing process approaches for development of models or IT systems. Additionally research results and ideas from several different fields e.g. from psychology, system theory, or Japanese manufacturing philosophies in the form of the Toyota production system, will be incorporated.

nCEA will combine the mentioned ideas and weave them into a pragmatic framework. An additional key idea derived in the thesis will be that of the 'personal process spaces' as a possible concept or tool in the modeling and handling of business processes. The concept of the personal process space is closely related to the concept of 'Subject-oriented Business Process Management' (S-BPM), which is a concept that fundamentally influences this thesis and also will be introduced.

Finally experimental settings to test nCEA are discussed, to evaluate and improve the approach, and test its usability. Afterwards a wrap-up of the presented ideas will be given and possible further research questions are discussed.

Introduction

The problem to be challenged by this thesis is rather simple: how to pragmatically and quickly improve existing business processes – with the focus on creating IT systems that supports said processes.

Supporting business process though is no easy task, because in addition to being complex themselves, it will be argued that they run within a system, a complex context. This context will inevitably be changed by introducing new ideas, systems, or tools – intentional or not. Not intended changes or side effects, though, are usually a hindrance and should be avoided.

The best way to do that is via the creation of a technologically independent model of the business process that will serve as the central piece for an IT-System (technology dependent). Creation should not be done alone by one person in some isolated chamber, but together with the people involved in the business process. And it should be done in order to not only elicit knowledge from them, but also to make them understand and accept, or even propose changes themselves.

Furthermore, only by interacting with the people involved – embracing a "go-and-see-for-yourself" [Like04] mentality – process managers will gain the knowledge and ability to make good decisions. As such, understanding, and before that gathering the right information is of course necessary.

The idea that individual humans and their interaction should be the focus for such a task is not a new idea and has been proposed in different contexts such as e.g. [Cijo08] - *"...we argue that innovation processes can only be managed using a bottom-up approach, that is if the individual also is a unit of analysis."* Or an interview about managing employees on a German business webpage [Fric10] claiming that "*nowadays rules [*for business processes] *need to be made together with the employees. For only they have the detailed knowledge of how the processes actually work. ".* Or it can be found simply in the first statement of the infamous Agile Manifesto with: *"Individuals and interactions over processes and tools"* [Beck01] , to name a few examples.

Success in a business project context is usually not measured in how much individual people were involved. It is measured in whether a project was executed in reasonable time, without interrupting business itself, how expensive it was, how many errors occurred during adoption, or which cost could be saved by changes. That list could be extended, but as it is it already indicates constraints that might be opposing the adoption of earlier stated ideas.

This thesis proposes a pragmatic agile approach that allows for achieving those goals. The approach is intended for model-centric system development environments that allow for very fast but still transparent transformation from model to a running business-process-supporting IT system.

The first section will introduce and discuss general concepts and ideas that have had an influence on this thesis and the general concept of 'natural exploration' is described.

The second section describes what actually will be done to use this concept in reality, including solving or avoiding problems that might arise while ‘exploring’

The third section incorporates the idea into the organization structure of SCRUM, to give them an agile, governing framework, required to not get lost in the jungle with all the ideas.

In the fourth section, an experimental setting is presented that examines how the concept of nCEA can be tested in a controlled environment, in order to assert is effectiveness and gain valuable insight into dimensioning or actual practicability of the approach.

Finally in section five, a reflection on the thesis and further outlook is given. There, the influence of nCEA and further research that can be done following this thesis is discussed.

1. What is a Process, What is a Model, and What is all this good for?

1.1. Ground Work and Basic Assumptions

Business Processes

Business processes do exists! In every organization there are people doing something to achieve the goals that organization exists for. For a regular business, this, of course, means to serve a customer in order to achieve a profit, whilst e.g. for public administration organization, this might be the processing of an application form. The term “Business Process” is usually used to describe or sum up an (usually often repeated) interconnected sequence - or web - of tasks and choices that together produce or lead to a certain outcome or service, which in turn provides (or should provide) a benefit for a customer or applicant. These tasks and choices happen partly parallel, partly depending on each other, and are executed in a complex interaction of several people and also machines.

Now in our fast changing world, continuous adoption to new circumstances and improvements are almost mandatory for any business to survive. Consequently the processes of any business[1] also must change[2], must become “better”[3].

Apart from measures to change processes structures (e.g. varying order of task, or limiting choices within the process flow), this can be achieved by supporting the execution of business processes with IT-tools. These tools can (or should) monitor and control work flow, or automate parts of a process. Both approaches (changing structure or creating supporting tools – or both) require a profound understanding of processes in order to be planned correctly. This must be an understanding of the static structure a process might have, as well as its dynamic behavior while being executed.

So it is not only the *tasks* and *choices* in a business process, or their order that need to be understood and to be considered, but also their context. A process does not “run” in a clean laboratory environment, but in the real world, where it is encased in a complex web of influence and/or restrictive factors, pre- and post-conditions and requirements that influence behavior of tasks and contributors factors. Factors that in their multitude and cross-interaction are usually hard to describe, but not to list.

These “side factors” may be[4]:

- Individual skills and skill levels
- Inter human relations

---

[1] For public administration the survival argument might not be true, but changes happen here often as well and need to be adapted to.

[2] Change as the most basic value-free description for: adaption, improvement, evolution, development, etc.

[3] “Better” *for a business process* does in general mean to minimize resources expenditures (concepts such as money, time, or opportunity costs due to minimized errors/better quality) used to fulfill the purpose of the business process.

[4] An in depth explanation of how these factors might influence events in a business context - either directly or by being a factor of consideration - will not be given here. It should, however, be fairly obvious that they do, and rather intuitive in which ways.

- Tools and techniques used (e.g. already existing IT-systems, or personal work flows)
- Organization(s) (in which a *Business Process is run*):
- (Sub-) Cultures / Languages
- Costs & Risk
- Ethics
- Laws, norms, regulations, rules

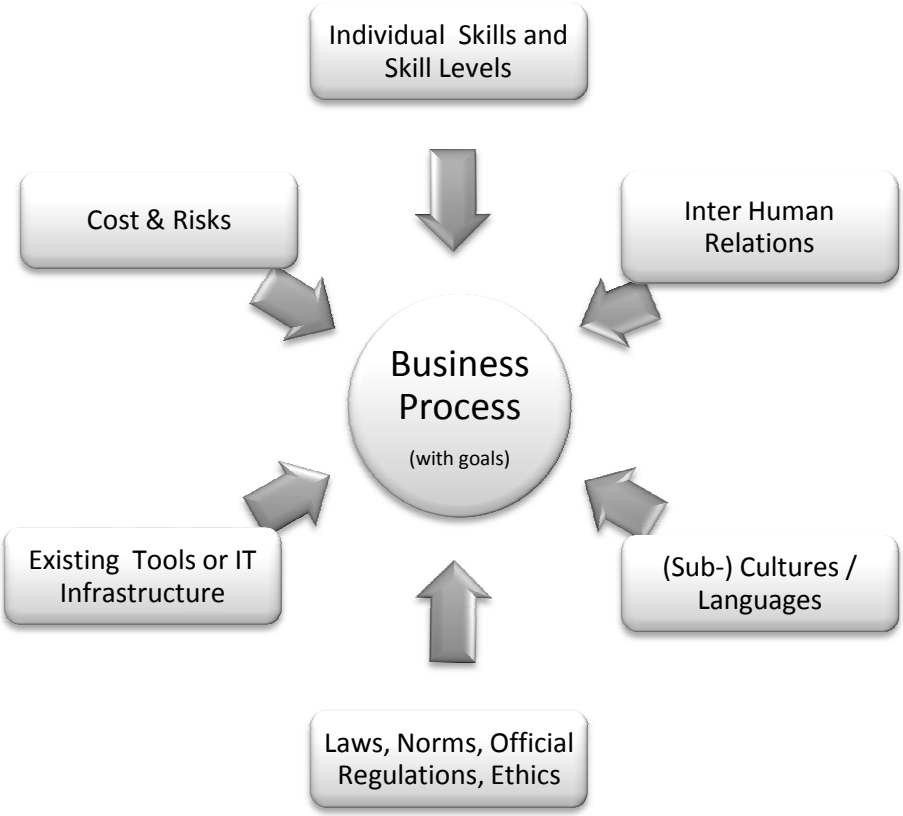


Figure 1: Context of a business process

While process steps may be changed 'relatively'[5] easy and fast, this context of a process may not or only slowly[6] be changeable[7]. Yet any measure that 'changes' a process will always not only have direct but also indirect consequences within this

[5] There usually are no easy changes

[6] E.g. company structures and organization may only be changed once a year, while process may need to be changed within 3 months. Equally hiring the necessary new, yet suitable employees might take quite some time. If an expensive SAP IT system has been in operation for 15 years it is unlikely this will change very fast even though it might be not the best solution for a particular task.

[7] This must not necessarily be a direct change, but already the introduction monitoring or control *structures* will have effects in one way or another in the context. In a strange way, the quantum physics law that you cannot perceive a system without changing it also applies in this case (Heisenberg uncertainty principle).

complex system and cannot be considered in isolation. Still, changes will and must be considered, because doing nothing is no option. But it is a form of art (and the goal of this thesis) not to work against the context but naturally with and within it to improve a business process. This especially means working with the people and also giving the people the opportunity to work on the process themselves.

This is by far not an easy task, as already larger processes – with many involved parties, task and optional paths (depending on choices made within the process) – can get relatively complex and hard to handle[8], even without considering the surrounding context or conditions.

To be exact, this surrounding context differs from individual to individual that is involved in a process, forming a kind of personal process space with only partly equal conditions around each involved person or 'processor' in a business process individually. Generalizing the maybe differing process spaces into a kind of monoculture is of course possible, but if not done carefully can harm the process e.g. by imposing unreasonable conditions or requirements on certain parts or processors.

Then way bother with the existing structures at all, and not simply raze them to the ground? "*It is wise to tame the power of as-is processes instead of trying to break them."* argued [Buch09] in conference speech, that described in detail the strong economical advantages of transformation approaches in contrast to complete reengineering.

The Problem of IT

Modern business processes usually involve information technology. Technology meant to support the operations of a business. But information technology is inert and changes the business environment itself, especially if large and old systems are involved.

"IT or software strategy, incarnated into bit and bytes," is a quote from a manager of an IT department. Yet at the same time, it controls every day operational work and as such needs to be adapted to operational requirements to really bring use to a business operation.

Figure 2 tries to depict this paradox of information technology. IT as the innermost facility is at the centre, and by expanding that theory of organizational/management science should have the fastest cycle time, but it does not.

[8] "handle" = understanding it, exchanging knowledge about it (talking) and deriving and actually implementing useful changes in the process

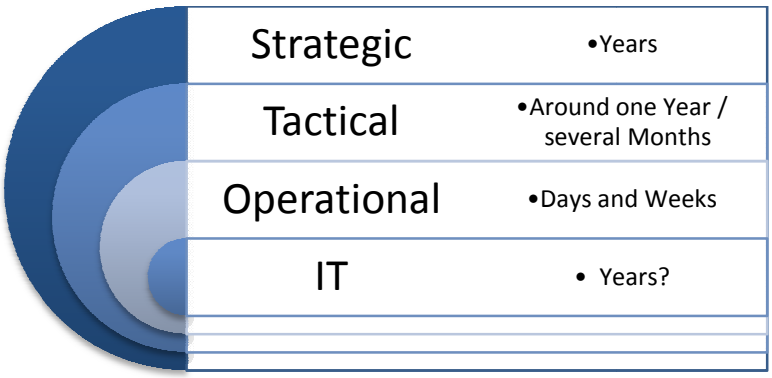


Figure 2: Discrepancy between IT and business strategies/decision levels

When a business strategy is derived for company, norms that govern everything else are set. Slow software development times do the same: setting norms; norms that after a while tend to be not sufficient for operational needs and become more a hindrance than a useful tool.

This must not always be the case, but it is certainly not uncommon and a part of the context – though a part that should be changed.

Software and IT should be something that can at least partly be adapted to operation needs, especially of complex business processes. Their use should not be the (partly) cause of complexity.

1.2. Modeling Business Processes

A Model of a Business Process

We humans are not well suited to really cope with complexity[9] and a business process in all its dynamic possibilities is usually described as being complex. To try to handle it, we must make use of models[10] that help us understand reality – to break down an aspect of reality into smaller parts and put them in certain order, and to give us something to actually see and "touch" and talk about. Furthermore, formalized models are also the only way to really tell our computers (finite state machines) about circumstances in real life, as they cannot by definition cope with the infinite possibilities of reality, nor are they able to even really record or comprehend it.

The first thing to be said about models – in this case, of course, models of business processes – is that they will vary in expressions, details, or formulations, depending on the planned use of the model after its creation. Specific examples for such uses are: as a tool for explaining functionality to others, as the basis for analysis[11] as the basis for IT system development, or – in more advanced versions – as the actual working core of

[9] From [Dörn09] & [Vest02] : Humans tend to think in linear causality-structures and also can perceive and learn about reality only in linear ways (one thing at a time). Here meant is complexity on multiple levels of processes and their structure, and also working within the complex context. Humans, of course, can overcome that weakness with their intelligence and learning abilities, by learning about reality and to create and use tools to overcome their limitations – such as described here. Intelligence and learning abilities in turn are dependent on our ability to communicate complex topics among each others.

[10] The term model is used here in a rather broad sense, but in context of this thesis "model" usually will imply a more specialist "process model"

[11] E.g. with a simulation tool

an IT-System (e.g. a workflow engine for processes). A main point of divergence – already implied before – is usually whether a model is intended for humans[12] or machines to be understood or to be read.[13] An ideal model that is easy to understand for both at the same time is usually not possible to the last degree, due to, e.g. the needs to extra-specify certain aspects for a machine.

Models are also/further limited by definition. They are artifacts, formalized expressions in one notation or another (nowadays usually digitally created with IT-tools) that try to describe a part of reality by focusing on certain aspects while leaving out, or abbreviating and summarizing many others. A model cannot cover every detail of reality and thus can never be completely "right". It can be valid in regards to the language or notation system that is used to create it. Otherwise it is more accurate to say that the details chosen to be expressed, or a certain way of modeling, are adequate or valid for the purpose of the model, but not wrong or right. This holds true as long as the expression is not clearly contradicting facts from reality, if the statements are *sound* – so to speak in the terms of formal logic arguing.

Admittedly it may be a grey area between whether a certain modeling expression is actually wrong or just not adequate for a certain purpose.[14] Though as reality and thus the context changes continuously, a model also can only be "adequate" for a certain time-frame and may need to be updated (or completely changed) in regular intervals in order to remain adequate[15].

As stated, what aspects and details a model consists of depends on the purpose it is made for. Respectively, it depends on what the persons creating the model actually know, or have learned about the matter and purpose that are to be modeled.

Models do not simply come into existence, they depend on the knowledge and understanding (about details AND purpose) of their creators. But knowledge about a complex system cannot be gained all at once, nor is it easily available.

Even if a formal documentation about a process prior to a more current modeling project does exist at all, it is rarely complete and at least not up-to-date. Other and up-to-date knowledge about what is really being done is usually only spread among the persons currently involved in the process. Thus this kind of knowledge needs to be extracted and collected (in a model), a task that takes time and has restrictions.

A Small Excursion into the Logic of Model Creation

Making a model is always a form of interpreting, describing and abstracting reality, and putting it into interconnected signs and statements. Thus a model can be interpreted as a *hypothesis*, an assumption or argument about the workings of reality. That argument should be constructed with *sound* expressions, which in turn are inferred from facts and

---

[12] Humans not necessarily fluent in a too formal modeling language – like being able to ad-hoc write down all diagram types defined by the UML standard

[13] In case of the former a model is often less abstract and has good graphical structure, but maybe "open-world", while the later usually requires formal correctness and stringent modeling – and usually a completed, closed-world-system.

[14] Another contradiction may be cases were compromises need to be made because modeling a certain aspect may (e.g. du to limitations of the notation) make modeling another aspect impossible to express the other aspect wrong.

[15] A requirement not all models, especially if entombed in large rigid IT-systems can provide.

knowledge of the creator of the model – his observations of the matter to be modeled (here specifically a business process).

Naturally for human beings, their observations of reality – complex as it is – are flawed, incomplete and can never be more. That is an undeniable fact. It is an idea already fundamentally expressed in the famous imprecisely translated statement of Socrates: I know that I know nothing.[16]

Consequently, formally speaking, a model can only be inferred or *induced* from the limited number of statements and observations the creators can make. The logical reasoning method of *deduction* is impossible by definition, because the system as a whole can never be observed and thus the *soundness* or *completeness* of arguments and statements cannot be guaranteed. Induced arguments, though, carry with them the possibility of falsification.

Unsurprisingly, the chance for new facts or observations to arise or to be made is high, as is the chance that these facts contradict or are otherwise incompatible with a model. As a consequence the *hypothesis* inherent in a model can no longer explain all aspects and thus needs to be adapted. Or rather, a new *hypothesis* needs to be induced; the model needs to be updated, if the new observations are not simply being ignored. Intuitively, this process should be easily discernible as a possibly never-ending cycle, since reality changes continuously, and there probably will always be an observation or other fact that could alter (or render more precisely) the *hypothesis*.

In the world of formal logics this kind of iterative procedure, *"…the whole process of generation, criticism, and possible acceptance of explanatory hypotheses*" [Jose94] , is usually described as abduction or abductive reasoning – at least in a broader sense, since the term itself and its precise meaning are still not 100% agreed on (as described e.g. in [Reic03]). Nevertheless the concept of Abduction has wide ranged foundation is used or referred to in several sciences especially in a sociological or social context (again [Reic03]).

Following this analogy, the goals of this thesis can also be described as proposing a pragmatic project/process approach that allows for and supports the abduction process in a business environment, for generating accurate hypothesis about a business process in form of a model.

But it is not that simple, because, as stated, building a model that matches reality is only part of the goal here. At the same time, quasi parallel, there is a second *hypothesis* that needs to be abducted – postulated and verified with data from reality – but within the same model: The model that is being created does express or produces the required results (e.g. it improves work via supporting the process with a workflow engine or IT-System). These two "*hypotheses"* are of course closely interrelated, but not 100% congruent.

Verification, or falsification and adaption of these hypotheses is not supposed to be the reasoning of a single individual. Because of the nature of business processes involving many person who have knowledge or can (and should) make observations about reality, what is necessary is a kind of collaborative abduction – team work. This is what

[16] The actual phrase is more closely stating that one can "know, that one does not know" [PISt97]

needs to be given organizational, operational form, and an environment within an already existing system that allows for finding or observing new information, sharing it, and together derive *hypotheses* and verify them against reality. This includes allowing the revision of previously made inaccurate statements if new facts have been learned about.

A Didactical Problem – Modeling from a Learning Perspective

"That brings the reader to the fact that learning is an important issue when dealing with complex problems. …. [And] *One possible way of learning about systems is to use models.*" [Hürl09] p.8-9. – which is exactly what has been stated so far.

But there is one problem: At school there is a teacher who sets a course of learning that guides students effectively through a complex topic to be discovered. The topic is usually well known and exactly pre-defined by the teacher. The knowledge to be taught is pre-organized, and pre-partitioned into a structure optimized for learning. This way it can be presented in an easy, mostly linear way of nicely scheduled, regular hourly units – making learning relatively easy. The didactical problem of optimizing the learning process is already solved there,[17] e.g. what to present to whom, in what way and order, to minimize time consumption or misconception probability.

This luxury, however, does not exist for learning in real life in general and in this learning task in particular. Knowledge is distributed among process participants, it is incomplete, it is unstructured, and it lacks predetermined borders. And there is no single person who has or even can grasp every aspect of a business process at once, nor is there a preset ideal course for learning topics or aspects.

There is not even a predefined teacher/learner role in such a task. Of course knowledge about the process needs to be transferred to the model creators, but also intention, methodology, consequences, and ideas need to be transferred to the 'other side'.

[Lien99] p. 324 reminds and proposes that: "*Even the older can learn from the younger, the grown up from the child, the superior from the subordinate. But one should not 'teach' or 'impart' something, but both could approach the necessary economical knowledge to be gained in rational discourse."* However he is only talking about two people or the elementary relationship between two persons in such a joint-learning-teaching-process, while such a discourse in a business process context usually involves a larger group of people.

The consequence of these circumstances: gaining information will always be a process over time, a collaborative, joint-learning-teaching process. There will be double information, misunderstandings, and contradicting information, to name the most common appearances. The participants need to discover and build the learning course together. And they need to be willing to go back and reargue about older matters if certain doubts or questions arise later on, because it is rather unlikely that every aspect can be covered at the right time in the right way on the first attempt.

[17] It is at least a feasible solution, since it is impossible to learn about complex matters in completely linear manner without looping or reexamine certain aspects.

Top-Down vs. Bottom-Up

Top-down and bottom-up are two strategies or concepts of information processing or ordering knowledge. They can be applied to describe or distinguish processes or approaches in computer science, management, or other fields. Strategies to create models of business processes are among them.

They (especially top-down) are based on the assumption that an intrinsic hierarchy within a system does exist. For a model that means a hierarchy where units of the system to be modeled are described in finer details on lower tiers of that hierarchy while on higher tiers such details are summarized to “make room” for other aspects of the system.

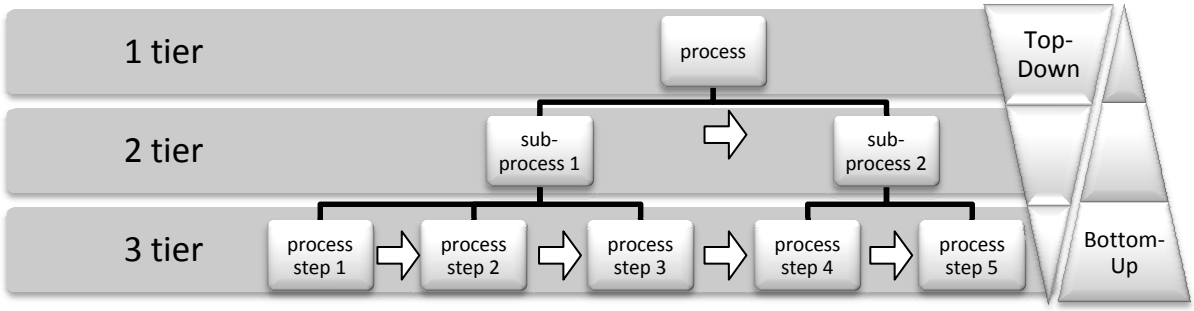


Figure 3: Exemplary depiction of a process hierarchy

Pure top-down creation of a model would mean that (according the systematic of Figure 3) first the process on the first tier is modeled (described), then the sub-processes of the 2nd tier are defined, and finally the individual process steps are incorporated into the existing model of the higher tiers.

A bottom-up approach would first model all the individual steps that are actually taking place, and afterwards try to aggregate them into higher-level process descriptions.

Both approaches come with certain limitations, especially if applied to large scale processes:

Bottom-up approaches notoriously lack an overview or a general ‘bigger picture’ and thus come with the danger of losing oneself in too much detail or doing much more than is reasonable, due to the lack of scope, not-defined borders of the process, or nonexistence of a strategy. Another subsequent danger lies in not having a reason to ask ‘why’ certain process steps happen or if they are necessary at all.

Analyzing a business process top-down does not include those problems, because it is more systematic and starts with predefining the general setting and strategies and overall goals. This may be useful in situations where processes and organizations to run the process can be designed in all aspects from scratch. But it is not practical in situations where existing structures need to be understood or incorporated. In the end, IT-systems need to support what is done at the bottom at the operational level, where the actual work is happening.[18] Top-down approaches have the danger to end up modeling something apart from reality. Models take time to build, and in larger systems

[18] They cannot support something that simply is not happening or in completely different ways than the system was designed for – that way they would only be a hindrance that is ignored most of the time.

by the time a model has reached a state where the lower level tiers can be modeled,[19] it may get hard to match the lower level process steps to a higher level process. In the worst case, the assumptions under which the first tiers were modeled (summarizing what was going-on on lower tiers) were not accurate, or even wrong from the beginning. Or because reality simply may have moved on and several steps on the lower tiers needed to be modified, because e.g. requirement does not exist anymore, organizations were restructured, or especially new steps or task were necessary; all problems that do not fit well with the top-down creation of models. In such cases, a rigid formal top-down model structure can make it hard to modify a process model and subsequently also the IT-system that is planned to support the process[20].

Another possible problem of top-down approaches is that because personal actual working on what in the model is described as lower tier, they are more rarely involved in the creation processes or only at the end. Consequently they tend to feel ignored and as such are more likely to not support (if not sabotage) changes or new tools.

Yet another danger of top-down modeling: especially with modeling paradigms that have rigid formalism in regards to process hierarchies such as ARIS, much time will be spent on agreeing what kind of modeling detail and action is actually appropriate on what level, or at least how details are interpreted and then assigned to a level. This leads to Discussions and work that are of course necessary for the sake of model correctness, but not necessarily efficient or productive for the overall effectiveness of the project or improving the process.

Overall, no pure top-down or bottom-up approach can be the solution (or is even possible). Instead a kind of synthesis is required – like building a bridge from two ends in order to get a final result that can be stressed in both directions. As any project or work needs a frame or starting point, determining a rough concept from a top-down perspective is pretty much mandatory. But after setting a kind of larger process-box in the model, it would be more pragmatic to fill out the content of the box bit by bit as is necessary for the purpose of the model using a bottom-up approach and by involving the person who will actually have to work with the results. [21] Adjusting model and details later – ideally at any point of time – is also something that should be possible in order for models to be useful.

1.3. A few Ideas to Combine: Existing Concepts that are Part of this Work

Models and modeling of business processes, though, are not the only aspect of the thesis. The circle of domains and concepts that possibly affect or are connected to the

[19] A task that may easily take six to twelve months with larger process using e.g. ARIS modeling strategy according to statements of interviewed process modeling experts (see Appendix A – Expert Interview Summaries)

[20] This naturally is bad if changes at the bottom are required because of new administrative regulations, new ideas, or new products. As a reminder: processes and support tools should in the first place exist to achieve a goal, not to fit into an organizational category.

[21] An interview with Dr. Albert Fleischmann (as well as others) was the main source for this section. Again see Appendix A – Expert Interview Summaries

topic is more than huge. This section will present more concepts and ideas from various backgrounds that have had an influence on, or are incorporated otherwise into one of the thoughts and ideas of this thesis. The ingredients of this approach so to speak.

The concepts presented here were chosen based on the evaluation of personal experience and evaluation of several expert interviews that were conducted as research for this thesis. The interviewed experts each brought up partly the same but also very differing concerns, ideas, and subsequently evaluations of how good certain aspects worked or not.

Without the proof of exhaustiveness the concepts introduced here represent those concerns and ideas. Larger parts of the approach are based on or derived from them.

Complexity

The terms ‘complexity’ or ‘complex problem’ or ‘system’ have been used multiple times so far (and will continue to be used) without having been defined more clearly up to now.

The word complexity itself is used often and in rather different contexts. Though, since the thesis is concerned mainly with human interactions, definitions from the field of psychology will be the most adequate.

Described here are attributes of complex systems or problems that have been identified to cause problems for humans to understand and reasonably influence or solve[22] them, and that we consequently must learn to handle, organize, or cope with.

(High) Number of Variables: Originally seen as the singular attribute to measure complexity. The more factors/switches/people to control or monitor (manage) the harder it is for humans to keep an overview and decide rational.

(High) Degree of interdependence: A system has a high degree of interdependence if the aforementioned variables are (inter)connected and cross-linked in many ways that can cause side-effects or consequences, which should be known.

Self-perpetuating dynamics: A common attribute of complex systems, caused usually by the interdependence of variables. Consequently, such systems have a tendency to change even without outside factors or only limited interaction.

Lack of transparency: Expresses the problem that usually not all involved variables or interdependencies can easily be perceived or are known and as such there usually is no complete understanding possible.

Multiple objectives (Polytelie): In contrast to ‘normal’ or complicated problems (e.g. solving a mathematical equation), with complex problems there usually is no single objective but multiple and often contradicting goals (a classical example would be the economic problem of simultaneous turn-over and profit increase)

For most of these six attributes, it should be obvious that they also apply in the context of analyzing business process and supporting or controlling them with IT systems – if

[22] In reference to [Funk92]

not directly in the formal process, then often indirectly via interaction or the unseen connection of the individual process spaces. This refers to hardly discernible interconnections on personal or semi-formal level that are also hard to elicit and even more problematic to record or note down. Sometimes, the indirect influence might even be stronger than the direct ones: e.g. even though mission statements and ideas proposed by corporate propaganda often claim the goals that should be valid for any member of a business organization, in reality every employee has his own goals or objectives in his personal process space[23].

Six Errors to be made during the Handling of Complex Systems

As the section before – that was defining complexity as something that humans have problems with – this section is also derived from the field of psychology, from the works of Dietrich Dörner [Dörn09] The German psychologist conducted several experiments to investigate human interaction, learning and understanding, and behavior while controlling complex systems.

In his book "The Logic of Failure" he list six classical errors or mistakes – caused by the factors listed in the previous section – that can be made while interacting with complex systems and that should consequently be avoided or tuned down if possible. For this thesis this relevant insofar that – to remind the reader – every action taken or policy employed to control, model, or improve a business process is always an interaction with the system that is executing the process. The system itself though is at the same time the object of learning and the context environment of such a project.

First Mistake: Wrong goals or focusing on singularly topics. Instead of trying to find ways to improve performance of a system as a whole, certain issues and measurements are focused on exclusively, while others are completely ignored as unimportant. As interconnectivity and interactive, time-variant feedbacks are the usual characteristic of complex system this might cause a kind of "repairmen" behavior – fixing one issue at a time while several other 'holes' or problems arise at other points due to ignoring other problems after the 'first fix'. A holistic view and proceeding is a better approach.

Second Mistake: Analyzing data in a linear and not interconnected fashion. Parts of any complex system are usually connected in various non linear, even feedback-looped, ways. Even if data for each unit is available, it is a mistake to not think about how and how strong the units are connected and affect each other. It is useless to collect large amounts of data if the data is not connected and placed correctly.

Third Mistake: Irreversible decision on scope and focus. Even if issues and taken to be taken care of are correctly identified in the beginning of e.g. a project, it is a common human error not to go back and reevaluate the set policies. Even in context of new information people tend to stick with the first ideas and actions to handle certain issues because they were successful in the beginning. This might continue despite possible signs showing directly that the effectiveness is decreasing. The ability to take back

---

[23] Consequently linking the goals of different process space (e.g. – not rating the single performance of individuals, but instead of groups that have cooperated) might be a strategy → reduction of complexity via uniting motives.

previous decisions and thus maybe – even indirectly – admitting mistakes is a key to the solution of this mistake.

Fourth Mistake: Not regarding side effects. In complex systems, actions will have side effects. If not regarded and at least tried to be estimated they will cause havoc in any plan. Uncovering or eliciting possible side effects – unplanned non-linear consequences of action – is thus as necessary as it is hard, because in order to find out the consequences in "what-if" scenarios, the "what" needs to be considered and not dismissed as unnecessary. Additionally, planning and organizing should always allow the possibility that something has been overlooked, and needs to be corrected.

Fifth Mistake: Tendency to oversteer. If no instant or almost-real-time, measurable reaction to controlling actions can be perceived by (impatient) decision makers, they tend to augment the measures taken, which, especially if done repeatedly, will usually cause more input resources to be used to create more output than necessary. Smaller, incremental steps that allow for evaluating consequences (especially under the circumstances mentioned with mistake three and four) are a possible remedy to this mistake.

Sixth Mistake: (ab-)use of authoritarian or dictatorial actions. This is caused by the – usually incorrect – belief to have completely understood every aspect of system and also having the de-facto power to control and change a system at will. It usually causes the person possessing them to decide issues in large scale all at once "in order to get things done". Considering the nature of complex systems, it is rather likely that single, especially large scale, policies or measures – initiated by a single human individual and not considering other opinions – will lead to unwanted, unfavorable results. Only because the theoretical power to decide on issues does exist, that does not mean it can be used wisely. 'Lucky guesses' in previous situations may lead to the idea that this still is the only way to do things.

System Thinking and Modeling

Back to modeling of business processes for a bit:

"System thinking, in contrast [to standard approaches in research that focus on linear causalities and subsystems, ed.]*, focuses on how the subject of interest interact with the other elements of the system – a set of elements who interact to produce a certain behavior – within a whole system."* [Hürl09] P. 59.

And: *"System thinking is the ability to see things as a whole. It combines the art of seeing interconnections and the science of explaining complexity. A central principle of system thinking is that a system is the interaction of its parts, rather than the sum of the individual parts."* [Hürl09] P. 75.

Two works shall be introduced that deal with the task of modeling complex topics as a tool that helps coping with complexity of real life. Both more or less incorporate or even call directly for a system thinking approach as the basis for dealing with and modeling complex issues: One by [LiST07] , and other is Frederic Vester's Sensitivity Model [Vest02] .

The work of [LiST07] is simply a short paper on how to use the so called 'Problem Articulation Method' to *"Assist Planning and Management of Complex Projects"*. It can

be seen as an epitome for the basic procedure to employ (rather rough) system thinking – in their case to describe the structure of a complex project of the European Space Agency.

The basic 'ingredients' are simple: The first step is to identify units and subunits of a system that is to be described. These units divide the project into smaller, manageable parts. They are described as "*collections of organized activities performed by people or automata to achieve as set of objectives*" (p. 5). They are distinguished as either 'object'-systems that seem to describe matter entities or 'service'-systems, describing functions that serve the objects. More detail about how these units in the system are defined or a good method for organizing the process to 'come up' with the definition is not given.

In a second step, real persons (in an organization) or stakeholders, involved with the units beforehand, are identified and mapped to those units. Finally the technique of constructing a "collateral structuring model" is used to visualize an integrated view. Figure 4 gives an example of what such a model would look like. The same figure also shows that the basic organizational principles of the units or the ranges that are covered with one such unit vary widely. To give an example: one such service-system unit is representing the concrete mechanical task of 'dismantling' the systems on earth once their use is over. Another unit, on the same level of the model – which is implying a certain equality of the system unit – is denounced 'Environment'. From the description in the figure it seems to include many different political and science institutions and organizational concepts.

Consequently, without having detailed knowledge about every aspect of the model, it really comprehending the model seems to be rather difficult. This must not necessarily mean that it is a bad concept, only that comprehension of the model-units is hard for non-involved persons, hindering the incorporation of additional external thoughts and ideas. This could be considered a drawback.

As has already been said, the paper leaves out details on the actual process of deriving those units, or the stakeholder and simply refers to: "*The analysis of each unit and its subunits….[*as a*]…recursive process." (*p. 5). There is no detail given on the persons involved, or actual manner. And at least from the text it can be deducted that the project structure will stay rigid once it has been modeled, not allowing for further changes once the project, that was planned using this method, is actually running.[24]

[24] Which might be a grave mistake - in reference to Dörner (See 'Six Errors to be made during the Handling of Complex Systems' section)

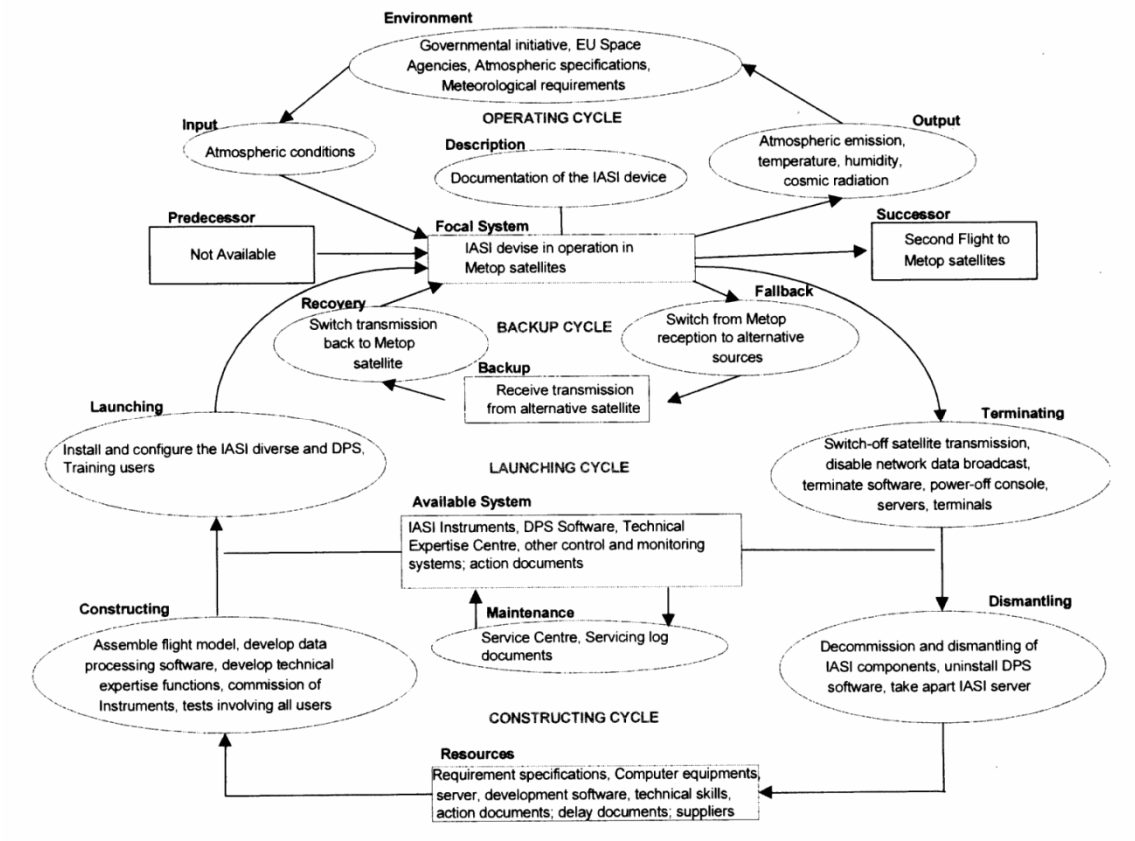


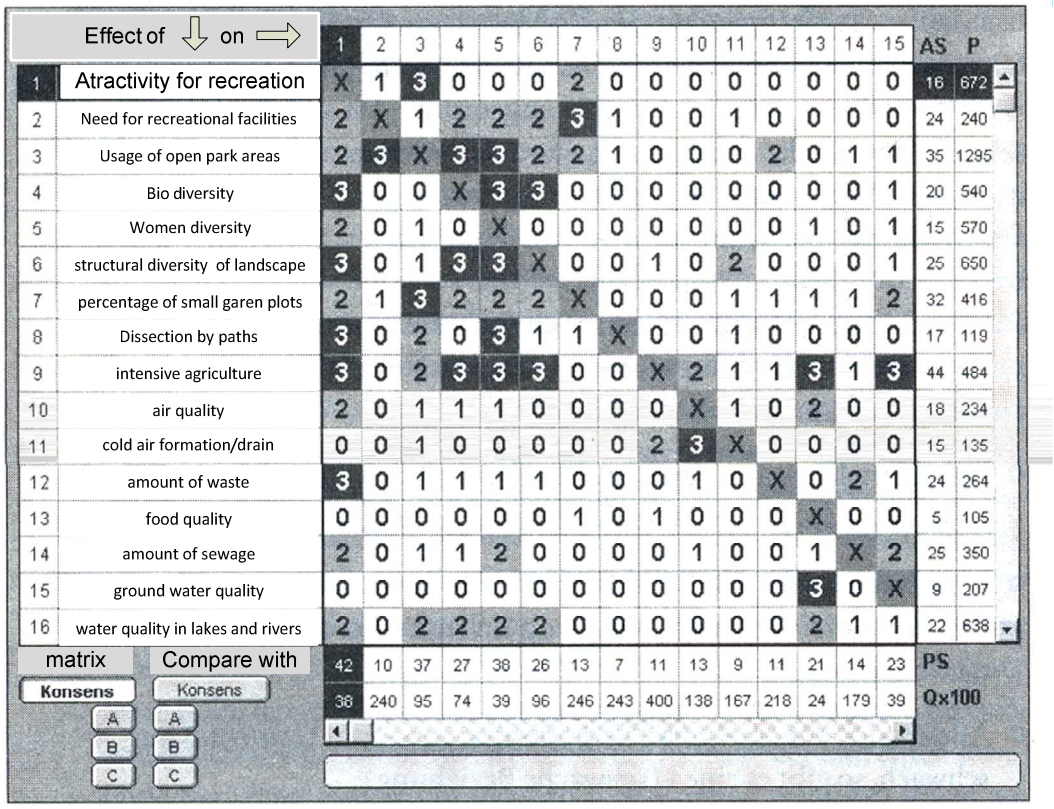
Figure 4: collateral structuring model of an ESA project from [LiST07] p. 10. Rectangles represent so-called 'object'-systems while ellipses are defined as 'service'-systems

In contrast to the project-management-focus of the previous paper, Vester's approach [Vest02] is much broader minded and meant to be used on a wide variety of domains. Vester also gives more detail on the process of how to construct a model of a complex system itself.

The first concept set by Vester is the surrounding or working environment in which a model is constructed and by whom. It is supposed to be a group of people from several different circles (from all would be best) of possible stakeholders to the system that is going to be the subject of examination with the model in order to get the greatest variety of input. They form a team over the whole duration of a modeling session that can last several days.

The units that the model of a complex system is composed of and that interact with each other are simply called variables. They should roughly cover several areas or aspects of life, and are limited in number from a minimum of 21 to a maximum of 40 in order to keep the system and the modeling process in a manageable dimension. The process of coming up with and agreeing on which variable to choose and to derive a common understanding is a process of social interaction. A quite important process that heavily comprises group discussions involving all participants.

Based on the elicited (and agreed-on) variables cross-impact-matrix will be derived. That matrix depicts how much these variables in the system influence each other (in scale from 0 – no interaction or influence – to 3 meaning strong dependence)[25]



Effect of ⇩ on ⇨		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AS	P
1	Atractivity for recreation	X	1	3	0	0	0	2	0	0	0	0	0	0	0	0	16	672
2	Need for recreational facilities	2	X	1	2	2	2	3	1	0	0	1	0	0	0	0	24	240
3	Usage of open park areas	2	3	X	3	3	2	2	1	0	0	0	2	0	1	1	35	1295
4	Bio diversity	3	0	0	X	3	3	0	0	0	0	0	0	0	0	1	20	540
5	Women diversity	2	0	1	0	X	0	0	0	0	0	0	0	1	0	1	15	570
6	structural diversity of landscape	3	0	1	3	3	X	0	0	1	0	2	0	0	0	1	25	650
7	percentage of small garen plots	2	1	3	2	2	2	X	0	0	0	1	1	1	1	2	32	416
8	Dissection by paths	3	0	2	0	3	1	1	X	0	0	1	0	0	0	0	17	119
9	intensive agriculture	3	0	2	3	3	3	0	0	X	2	1	1	3	1	3	44	484
10	air quality	2	0	1	1	1	0	0	0	0	X	1	0	2	0	0	18	234
11	cold air formation/drain	0	0	1	0	0	0	0	0	2	3	X	0	0	0	0	15	135
12	amount of waste	3	0	1	1	1	1	0	0	0	1	0	X	0	2	1	24	264
13	food quality	0	0	0	0	0	0	1	0	1	0	0	0	X	0	0	5	105
14	amount of sewage	2	0	1	1	2	0	0	0	0	1	0	0	1	X	2	25	350
15	ground water quality	0	0	0	0	0	0	0	0	0	0	0	0	3	0	X	9	207
16	water quality in lakes and rivers	2	0	2	2	2	2	0	0	0	0	0	0	2	1	1	22	638
matrix	Compare with	42	10	37	27	38	26	13	7	11	13	9	11	21	14	23	PS	
Konsens	Konsens	38	240	95	74	39	96	246	243	400	138	167	218	24	179	39	Qx100	
A	A																	
B	B																	
C	C																	

Figure 5: Example of Influence or Cross-Impact Matrix from [Vest02]

Based on this matrix, several thinking approaches and methods are applied to analyze and discussed several aspects of the system (including e.g. construction of a visual model of cross-linked units, with their dynamic interaction - Figure 6). Overall Vester divides the creation process into nine working steps that are to make sure that the system model is appropriate and allows for deriving transparent results and understanding from the model itself, as well as the model building process. Additionally [Hürl09] p. 90, analyzing Vester's model, advises: "*Theses nine steps are again not just sequential, but iterative in the sense that results of any step can yield insights that lead to revisions in any earlier step.*"

One important point is that all these creation-methods need to be dealt with together in group work with all participants present. The goal is to create acceptance of the model with participants, which is the perquisite for acceptance of decisions and actions derived from planning approach based on the model.

[25] Usually several versions of a cross-impact matrix by several sub-groups of the model creating team are derived. These matrices then are united in a group discussion to form a consensus-matrix that reflects the opinions on cross-influence of the whole group. This consensus finding is also integral part of the model creation process

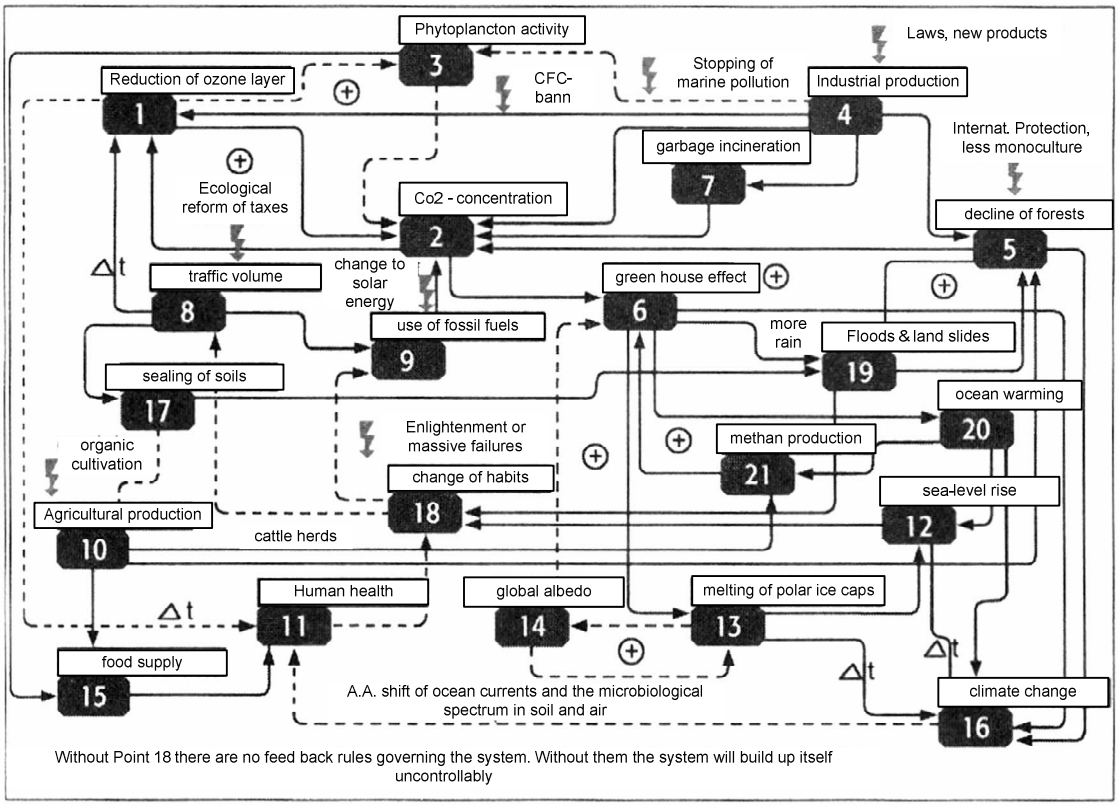


Figure 6: Example for cross-linked interaction model in [Vest02] (not related to the previous figure)

This brief introduction to both examples must be sufficient. But there are a few important aspects or points on model creation of complex circumstances that can be taken from them (and that match with other none-cited work on this topic):

- The process of modeling always should involve all kinds of possible stakeholders, if not in person than at least in consideration.
- Models do indeed grow, and structuring and organizing their development process is important. The people involved must at all steps understand what is done and why (based on Vester).
- The standard way or best practice[26] of modeling complex circumstances is first to identify units, separable smaller parts of the system, and then try to discover and represent how they are connected and play together – system thinking. The nature and concept of units in a system, though, is highly dependent on the underling modeling concept.
- Model creation is an iterative process and may be recursive.

A final thought on system-thinking and bottom-up: System-thinking does not necessarily implicate a top-down approach, even though both presented works have strong tendencies towards that methodology. System-thinking and bottom-up are easily combined if the units at the bottom of the system and especially the question of how they connect to their fellow units are regarded and the whole system is discovered bit by bit along these system-connections at the bottom. In a bottom-up system-thinking approach it would be greatly helpful if units at the bottom are easy to identify and define

[26] Simply because many people do so.

without having to discuss a long time on delimiting and distinguishing them. The human being (and its personal process space) can be a natural, intuitive, and pragmatic solution to that requirement.

Deming Cycle

The Deming Cycle presented in [Demi00] is a simple process model or concept, that describes a continuous (feedback) cycle of four steps representing the most elemental actions of an approach to improve any kind of (business) processes. The concept is based on scientific methods and comes from the domain of statistical quality control in manufacturing processes.

The four steps described are:

Plan: Set a scope and plan the changes of a process in order to achieve a goal or objective which also needs to be established

Do: Implement the changes.

Check: Measure the effects of the changes and compare them to similar measures before the change.

Act: Analyze and evaluate the results and identify the cause for deviations from planed or expected results. Adjust scope accordingly .

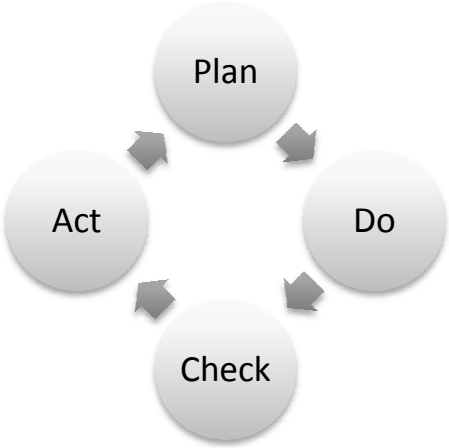


Figure 7: Schematic of the Deming Cycle

This, of course, is only an ideal sequence, describing what should happen. It incorporates and represents the elemental idea of continuous and incremental improvement, including the implication that changes or ideas that do not work out need to be taken back or reformed.[27]

[27] The basic concept is also akin to (or could be interpreted as an implementation of the idea of) the process of abductive reasoning briefly explained earlier: Plan: the hypothesis, DO: experiments or collect data, CHECK: and evaluate the results according to the hypothesis, ACT: if the hypothesis is (partly) falsified and needs to be adapted → come up with a new PLAN.

Variations of this concept can be found incorporated in many other approaches and areas[28], sometimes derived directly from it, sometimes developed on own ideas or perceptions of reality. For the basic idea, it also does not matter whether it is applied for big projects and major changes, or only in smaller manners.

Implementing it in reality and complex situation though is not always very effective, as measuring of interconnected and dynamic effects can be an imprecise science and leaves much room for misinterpretations or misconceptions

Nevertheless, the wide popularity of the cyclic concept is at least somewhat proof to its effectiveness.[29]

genchi genbutus & kaizen

Genchi genbutus and *kaizen* are two basic concepts, among others, that are the corner-stones of the Toyota Production System (TPS)[30] – a system combining several procedures, techniques, or management concepts in order to continuously improve manufacturing processes. [Like04] himself calls it *"a system designed to provide the tools for people to continually improve their work."* P. 36 TPS is considered an epitome or at least precursor of lean manufacturing, management, and development approaches, and has been (at least partly) adopted or imitated by many other companies.

The idea behind the *kaizen* (改善) (English: Continuous Improvement) is basically identical to the concept of the Deming cycle. Within TPS *kaizen* is explicitly described as involving all levels of employees – with a strong emphasis on self organizing and team work – in order to control and improve quality and process performance. In contrast to authoritarian management style the improvement should come from within the organization, from all people involved themselves.

The overall goal[31] is to eliminate three types of waste that are not limited to production but appear in any kind of business process and reduce the effectiveness of said processes:

- *muda (*無駄, "useless") meaning e.g. unreasonable or unnecessary work steps in a process , which should be avoided.
- *muri* – (無 理 , "unreasonable") meaning work for process participants that overburdens them or is unreasonable and as such reducing the individual performance,
- *mura* (斑 – "unevenness") addressing the problems and costs that might arise if a system that is executing a process is running at an unsteady pace, meaning that it

[28] The DMAIC cycle of "Six Sigma"-approaches is one example for a variant of the concept. Another is the general cycle idea of agile software engineering approaches, such as SCRUM (see last section).
[29] E.g. Cyclic or iterative progression is explicitly part nowadays of nearly every process approach in software engineering away from linear approaches such as the infamous water-fall-model or the V-Model in earlier incarnations.
[30] From [Like04] but also much referred to in [Larm09]
[31] Not necessarily of *kaizen* activities only, but pretty much the whole idea behind TPS.

gets times of overwork and then again other times where there is nothing to do because of uneven workloads over time[32].

The last sentence also already implies (again) the concept, that the three types of waste are partly causes for each other and cannot be considered separately in context of the system that runs them. Here e.g. that *muda* can be a consequence of *muri.*

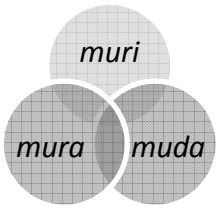


Figure 8: Interconnectivity of the three types of waste in TPS from [Likn07] p. 171

Even though the ideal implied here would be continuous self improvement without 'outside' interference, this is rather unlikely in real life and TPS does not even imply that because it never even considerers the idea of the absence of managers, their actions and advices and control. It rather embraces the idea that those people should constantly know what is actually going on in order to be able to advise their staff and make educated decisions.

This is expressed in the concept of *genchi genbutsu*[33] (現地現物 – "genuine place and products", usually referred to as "go and see" principle in English literature). Genchi genbutsu in the most basic description calls for any decision maker to go and see for themselves what is happening down on the shop floor at the actual working level. Not only to see, but to understand the steps the workers do and even be able to do those task themselves – to experience the real processes. The idea expressed in that philosophy is that every manager (who wants to improve something) needs an intuitive and practical understanding of the matter he is managing and working with; something that can only be learned by experiencing the actual conditions on-site, doing real action in contrast to only reading reports or listening to presentations.

As [Larm09] has shown, there are many more principles within the TPS – concepts like a fundamental respect for people and teamwork – that can and should easily be applied to the development or projects outside of car manufacturing. Here, presentation of only these two is sufficient, together with the emphasis that these concepts or ways of thinking are meant to be practiced by all personal involved – which in itself is also a principle of TPS.

The success of Toyota should at least partly legitimate the effectiveness of their TPS overall and the concepts of *kaizen* and "go and see" in particular, even though it is not without criticism and needs to be adopted and accepted in particular circumstances

[32] A typical example would be the "bullwhip effect" in logistical supply chains, causing uneven stock levels and order spikes across several stages of a supply chain.
[33] In [Like04] chapter 18 is dedicated to the go-and-see principle being the 12th of 14 key principles of TPS

Agile Development Approaches

In contrast to the car production that is the goal of Toyotas production system, software development is no industrial manufacturing process. Over the years the creation of programs has become more and more complicated, with software teams and lines of codes growing. Not to mention the rising level of complexity within the product.

The original common rigid software development practices were plan driven, with set-in-stone phases and phase deadlines, and completely preplanned and micromanaged schedules. Using such methods increasingly led to problems, delayed projects, suddenly unexpected cost increases, or simply bad software.

When creating or working with new, unknown technologies, pre-planning and determining everything a priori is illusionary. Giving orders top-down on something that did not yet exist, that nobody had an idea how it would work, or what problems might arise in between was and still is impossible, because only during the course of development many problems or new ideas surfaced (and they still do).

Consequently to be able to cope with these symptoms the mid 1990's saw the development of the so-called agile development approaches. The term agile was constitution in the agile manifest [Beck01] in 2001. The goal of these approaches was and is to provide a concept or framework that is suited to produce good results even in the uncertain situation as is the production.

There a lot of such development approaches, among them are 'eXtreme Programming' [Beck06] , the 'Unified Process' [Scot02] , or 'Scrum' [ScSu10] . Nevertheless they have certain things in common. The general spirit can be found in the already mentioned 'Agile Manifesto'

> ***Individuals and interactions*** *over processes and tools*
>
> ***Working software*** *over comprehensive documentation*
>
> ***Customer collaboration over*** *contract negotiation*
>
> ***Responding to change*** *over following a plan*
>
> ***[Beck01]***

The items on the right site of the sentences are still important; the priorities though do lie on the first parts of the statements. They function as a guideline a development project should adhere to in order to be successful and also not to frustrate its participants.

Additionally there are twelve principles behind agile (software) development (also on [Beck01])

- Customer satisfaction by rapid, continuous delivery of useful software
- Working software is delivered frequently (weeks rather than months)
- Working software is the principal measure of progress
- Even late changes in requirements are welcome
- Close, daily cooperation between business people and developers
- Face-to-face conversation is the best form of communication (co-location)

- Projects are built around motivated individuals, who should be trusted
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular adaptation to changing circumstances

The incarnations of the Agile Manifesto in real existing process-approaches vary in their strictness and prescriptive, as can be seen in Figure 9. The more an approach is pre-structured or prescriptive the more it is already specialized onto a certain field of development. Especially the 'Unified process' (formerly RUP) and to a certain extend eXtreme Programming (XP) are restricted to the field of software development. Less prescriptive approaches allow room for adoption to other domains. Especially Scrum can thus be used in wide range of contexts and ideas, according to [Larm09] .

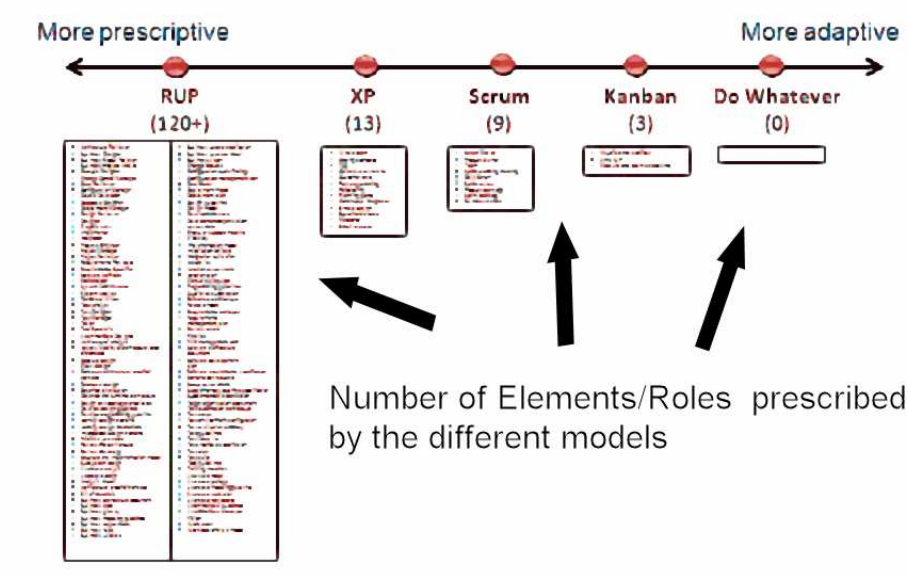


Figure 9: Schematic Comparison of software development approaches from [Knib09] p. 9

Most approaches have certain aspects (based on the 12 principles) in common. By that definition those common aspects can be interpreted as important factors for the success that the agile concept has[34]. They are:

- Time boxes (measured in weeks): Even if many things of agile methods are not rigid, the timeframe is – in order to give a project the guideline. During such a time box all the general activities of development. At the end should be a usable product
- Iterations and incremental development: A time box represent only one cycle of many that will repeat over the course of a development project. During each of these iteration the final product is more and more refined. Requirements can change as needed and the future increments will reflect those changes

[34] The proof to that is the seemingly growing amount of publications and success stories that can be found in literature nowadays. Furthermore there are surveys about success of agile practice, e.g. [Veri08] stating numbers of more than on in two people calling 90-100% of all their agile projects successful with only a small percentage of under 10% that were not at least partly successful with the methods

- Constant ‘customer’ interaction: there are people developing the product and people that want something developed. In order to constantly match the two sides many approach call for a constant participation of the people that something is developed for. This is usually in the form of a ‘Product Owner’ that decides in the interest on the customer on what to do.
- Small, self organizing teams that are coordinated and not directed:[35] the base principle here is that developers are told ‘what’ to do, not ‘how’ to do it. As micromanagement of people is simply not effective or impossible.

The original ideas for the manifesto and the agile development approaches was software development, as can be seen in some of the statements. nCEA – the process-approach constructed in this thesis though is not limited to software development, but rather incorporates that as part of a larger ‘business process development’ project.

Nevertheless there is a lot of common ground between software and process development: each is thinking-intensive work by a group of people that needs to be structured and organized. Thus it needs to be given a common ground, a framework to run in. Pre-planning is hard due to lack of exact knowledge what may come up during the project.

Taking a closer look at the proposed concepts of the agile manifesto (or the approaches) it could be said that they address at least parts of the problems or issues listed in the previous section, even though they do not explicitly name them. But it can be expected that they can be used to effectively manage those problems. Examples for this claim can be seen e.g. in the technique of pair programming or daily group meetings (in Scrum and EP) that have characteristics of joint-learning-teaching methods. Another example is the self-organization idea for teams that is akin to avoiding the sixth of Dörner’s mistakes: do not do any dictatorial actions (in the software development systems with the developers as the units to control). Or the incremental development idea, that is related to the *kaizen*/self improvement idea of the TPS.

Overall, to a certain amount of certainty it can be assessed, that any development work with uncertain outcomes or hard-to-plan aspects should incorporate at least some of the agile principles in order to be successful. For nCEA, ‘Scrum’[36] is adapted and used. It will be introduced in greater detail later on.

Of importance in this section was to show where certain ideas originated, what their basic content is, and how they are connected to the other parts of the thesis.

[35] (E.g. in Scrum) teams sizes of roughly seven people. For larger task several teams are grouped and coordinated.
[36] This is not unusual since Scrum has been adapted for general project management as is shown in [Schw04] or [Larm09]

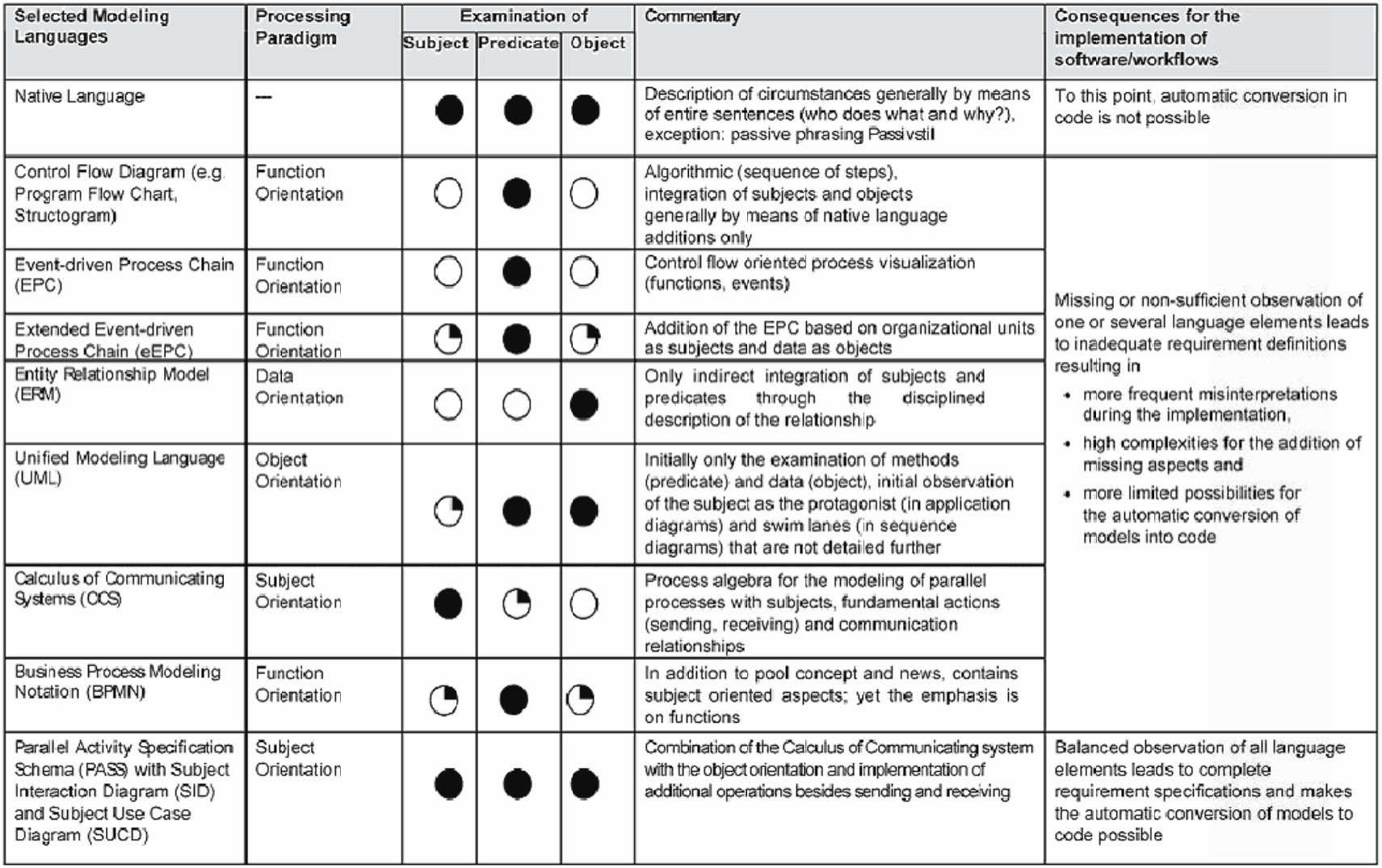
1.4. S-BPM Modeling

In general this thesis is meant to be used with any modeling notation and the concepts of the previous sections are not necessarily tied to anything in particular. Nevertheless, the original idea and still the main focus for this work were and are on providing a creation-process-approach for 'Subject-oriented Business Process Management' (S-BPM) that in turn has influenced the ideas of this thesis greatly and shall be briefly introduced.

Presented by [ScFlGi09] [37] the basic idea behind the approach is, that existing BPM modeling paradigms lack the freedom or power of expression of natural language. This is due to the circumstance that they are mostly concerned with the equivalents of grammatical verbs (procedures) and object (things to work with), and most often leave out or only marginally including the idea of subjects (who) that would be required to form a complete sentence in human language Figure 10, taken from the aforementioned paper, depicts that idea. Only with subject, verb, and object [38] a complete understandable sentence can be formed, and as such a modeling notation should be able to express them.

[37] An English version can be found at [ScFlGi10]

[38] The typical order in English (SVO) is of minor importance and might actually not be the most basic form. The form of subject, object , verb (as it can be found e.g. in Japanese) seem to be the more natural order which can also be deducted in Small Children phrases such as maybe "Mama, potty going" or such. A formal source could not be found for this claim, though it is based on a radio broadcast with a famous linguist on 'Deutschlandfunk' a few years back.



Selected Modeling Languages	Processing Paradigm	Examination of			Commentary	Consequences for the implementation of software/workflows
		Subject	Predicate	Object		
Native Language	—	●	●	●	Description of circumstances generally by means of entire sentences (who does what and why?), exception: passive phrasing Passivstil	To this point, automatic conversion in code is not possible
Control Flow Diagram (e.g. Program Flow Chart, Structogram)	Function Orientation	○	●	○	Algorithmic (sequence of steps), integration of subjects and objects generally by means of native language additions only	Missing or non-sufficient observation of one or several language elements leads to inadequate requirement definitions resulting in • more frequent misinterpretations during the implementation, • high complexities for the addition of missing aspects and • more limited possibilities for the automatic conversion of models into code
Event-driven Process Chain (EPC)	Function Orientation	○	●	○	Control flow oriented process visualization (functions, events)	
Extended Event-driven Process Chain (eEPC)	Function Orientation	◔	●	◔	Addition of the EPC based on organizational units as subjects and data as objects	
Entity Relationship Model (ERM)	Data Orientation	○	○	●	Only indirect integration of subjects and predicates through the disciplined description of the relationship	
Unified Modeling Language (UML)	Object Orientation	◔	●	●	Initially only the examination of methods (predicate) and data (object), initial observation of the subject as the protagonist (in application diagrams) and swim lanes (in sequence diagrams) that are not detailed further	
Calculus of Communicating Systems (CCS)	Subject Orientation	●	◔	○	Process algebra for the modeling of parallel processes with subjects, fundamental actions (sending, receiving) and communication relationships	
Business Process Modeling Notation (BPMN)	Function Orientation	◔	●	◔	In addition to pool concept and news, contains subject oriented aspects; yet the emphasis is on functions	
Parallel Activity Specification Schema (PASS) with Subject Interaction Diagram (SID) and Subject Use Case Diagram (SUCD)	Subject Orientation	●	●	●	Combination of the Calculus of Communicating system with the object orientation and implementation of additional operations besides sending and receiving	Balanced observation of all language elements leads to complete requirement specifications and makes the automatic conversion of models to code possible

Areas of emphasis in selected modeling languages and consequences for their implementation of software/workflows

Figure 10: Process modeling languages overview (from [ScFIGi10])

As a consequence, they present a process modeling paradigm called “Parallel Activity Specification Schemata” (PASS) that is composed into two separated modeling levels.

On the ‘upper’ level of the process level is the ‘subject interaction diagram’ defining process participants (subjects) and how they interact (via messages)

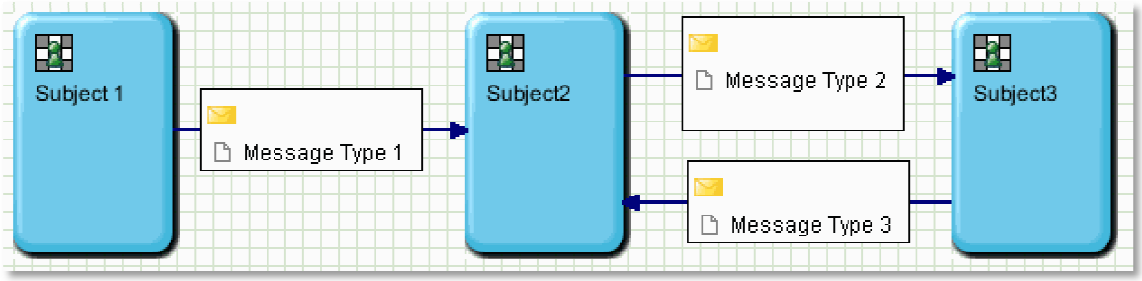


Figure 11: Example S-BPM subject interaction diagram

Then, on the ‘lower’ level, for each subject an individual work-flow can be defined in a “subject behavior diagram”.[39] Instead of an elaborate modeling notation such as BPEL or ARIS, the notation for S-BPM opts for a simply notation composed of only three possible states: send, receive, and function used to define interaction on a ‘personal’ level.

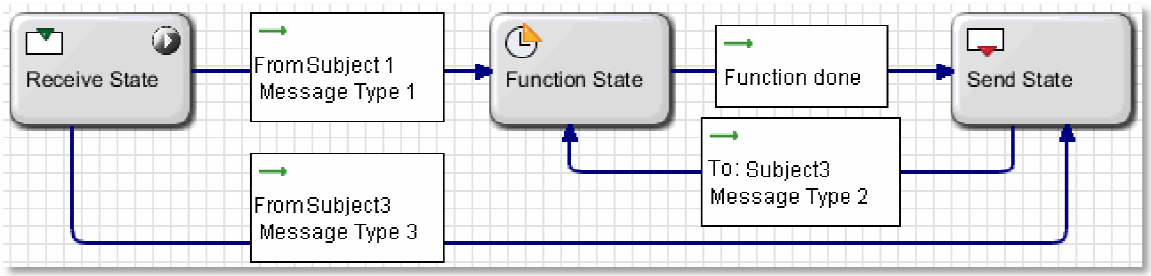


Figure 12: Example of S-BPM subject behavior diagram (within subject 2 of Figure 11)

The simplicity of the modeling notation, while still being Turing-complete,[40] helps to reduce confusion that is caused by overloading unschooled viewers with too semantic rich details and thus preventing understanding[41].

Thus, the modeling concepts main focus is on depicting complex communication patterns in strongly divided work processes, making it ideal for process in which communication, especially between people, is a major component of value creation. This holds true especially for the service sector.

At the moment S-BPM can be found imbued in the in the S-BPM Suite of Metasonic AG [Meta10]. Their suit includes a modeling tool, a validation tool that helps

[39] Another expression or description is: “within” each subject.
[40] According to a verbal statement by Dr. Fleischmann. Additionally, in a bachelor thesis done at KIT’s AIFB it was shown by Norbert Graef and Nils Tölle, that all of Wil van-der-Aalst’s 43 general workflow patterns (as presented in [AlHKB03]) can at least be reduced to a maximum of 25 different patterns in that notation.
[41] While being very expressive many, process modeling languages are hard to learn and tend to confuse people who are not familiar with such a notation.

demonstrating and playing through the models at hand, and a process engine to directly run the process models in a productive IT-environment.

Advantages of S-BPM Models

S-BPM based modeling of a business process offers several advantages, especially combined with ideas of exploring a process.

The paper of [ScFlGi09] itself claims improvements such as the ability for "real time"/fast modeling, (with the existing tools) the simple integration of existing services, or a greater degree of automation in the IT implementation.

But S-BPM modeling methodology has a further advantage, making it especially useful for the exploration of business processes and collaborative development of business process models: It is really easy and fast to understand for humans and there are several reasons for that.

First, to understand a given situation humans tend to have a certain internal algorithm, a set of questions they want to have answers to. The classical 'Five W Questions'[42] that every journalism student learns as the basics that he has to answer with each article, but that are also taught e.g. in police training, are the epitome of that: Who, What, Where, When, Why, (+ How?[43]). From personal experience, it can be deducted that in this case the order of the questions fits that natural algorithm, and S-BPM's modeling notation thus answers first the most important question for humans': who is actually involved[44]. Everything else (actions or reasons e.g.) is connected afterwards, to those persons. The reader of the model does not need to guess (or at least keep in mind) who is actually doing what.

Secondly, since subjects can usually easily be mapped to real existing persons or entities, the workflow within one subject matches the task order of single process participants pretty closely. Those process participants subsequently do not have to identify their tasks from a big tapestry depicting a whole process model. They can directly relate to and verify or correct workflows for themselves. They can easily identify their own process surrounding (or context) within the model, also making it easy to imagine the task of others who might be working simultaneously.[45]

Thirdly, for the first point to be true (comprehensibility) the model is – in contrast to other modeling notations – not required to be complete or drawn out in all details to

---

[42] The Five W page on Wikipedia [WiFi10] gives the fastest introduction to the concept including links and list of sources tracing the concept back in history – which most often also start with the "who" question first.

[43] Which does not count because obviously it starts with an 'h' – in German it would work out because 'how' in German is 'wie', thus forming 6 Ws for that language.

[44] Other modeling notations, of course, also include the concept of people, but on secondary, only additive level. Though while designing, most modelers will have 'some person or entity' in mind for any given task or decision in a business process. Similarly most other persons involved with the model creation will have at least a latent concept about the other participants in mind. If a non-knowledgeable person however has to work with the model without knowledge about actual distribution of tasks it will be much harder for them.

[45] Humans tend to describe and think in metaphors, or in "what they are used to". Describing a process in terms of processors as e.g. "boss", "coworker", etc. allows for easy and fast orientation.

make some sub-sections understandable. It can grow in detail for some areas, while in others it may not need (yet) to be finished. Modeling can start without a big picture (which may be often incorrect, or not adequate to fit all details from lower tier into it). This kind of big picture may be required in hard-defined top-down approaches. A subject can be completely modeled and verified without having more than stumps in the surrounding; stumps that in a task-oriented process-flow diagram would translate to maybe lots of unconnected units with lots of white areas in between where other, not yet involved subject do their tasks. Figure 13 gives an example what this might look like.

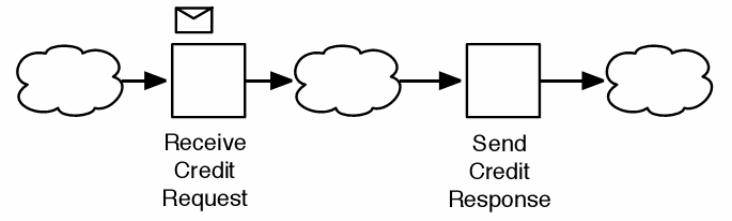


Figure 13: Abstract, incomplete non S-BPM structured workflow that leaves out many areas in between. (from [Puhl06])

S-BPM in contrast allows the model to grow unit by unit, like simply connecting pieces of a puzzle, instead of needing to draw the complete picture on an empty canvas yourself. Pieces, of course, still need to be crafted.

Fourthly (in the incarnation of the Metasonic Business Suite), by being able to easily bring the model directly into a running IT-system – even if only partly – S-BPM greatly helps in reducing semantic transaction losses. Semiotic transaction loss occurs when transferring the modeled idea of the real world, first to the sings or models of the business modeling domain (classical business process management), then the business model to the IT model domain (with technical models and requirements). With S-BPM there is only one transfer step. "*This [*the problem of semiotic transaction loss, ed.] *will consequently cause inconsistency between the evolution of the information system and the business change",* as [GaLiHa07] ,who have analyzed the semiotic side of the problem of system design, state it,.

Ultimately, the given advantages may not automatically lead to an improvement of model and IT-system quality, but it will make working and interacting with the involved process participants faster, more goal-oriented, and more efficient.

This hypothesis is based on the findings of [Hess82] , who – based on Dörner's work – investigated the behavior of humans in interaction with or in control of complex systems. Results from his research indicate that when faced with a complex task, giving the involved persons a familiar, easily understandable context structure[46] may not improve the overall quality of the results (but it does not worsen them), but greatly increases the motivation of the participants and speeds up the whole process.

[46] Meaning that the complex task to solve was encased in semantically rich, but also familiar, already partly known, and well explained scenario. So overall the task was made easier to understand and relate to.

Natural Context Separation – Core Advantage of S-BPM

Any modeling paradigm sooner or later runs into one problem: in real life business processes tend to be enormous, spanning many people, departments, and functions. To still be readable and function as help for people to understand and organize the complexity of reality, models sooner or later need to be split up and separated. Also, simply due to the time needed to create them, models of larger size are harder to handle, especially up to a point where they are covering everything intended and are still consistent and valid in regards to their statements about reality.

Another problem connected to that topic is the question of detail factor (or zoom level) within a model – or on certain levels of a model if there are explicit process hierarchies used as it is done within e.g. ARIS. The more people work on a model, the more work has to be done to coordinate them and make sure that the detail level of the process model is (at least) similar on all levels of the process.

The central advantage of S-BPM (responsible for the positive effects of the previous section) is that from the very beginning, it forces the modeler to break up and separate the whole process model into small, individually manageable parts (which is in essence not different from e.g. what is done in the two presented examples in the section about system thinking). Because of this person- or subject-centering of the models, the context is also split up into individual personal process spaces. This is not necessarily done explicitly in the model. But it is undoubtedly easy and intuitive to understand that the surrounding conditions may differ for separate persons because they are part of different departments, human-resource-hierarchy levels, educational backgrounds, or clearance-status.

This separation of model and context is not done on a whim, but definitively along existing units that are intuitive to understand, that are the natural units of the system (in contrast to being abstract or vaguely defined). Herein lies the difference to other system or unit dependent modeling approaches (e.g. the examples given in the 'System Thinking and Modeling' section). For practiced users of such modeling approaches, the definition and handling of the chosen units and description of their interaction will seem trivial. But for example in Vester's approach the choice and definitions of the units (variables) is an important task that takes time and requires intensive discussions. Likewise understanding the range, extent or dimension of Figure 4 on page 16 is rather hard without having been introduced extensively.

Along these units a whole process can be explored bit by bit and then modeled. To be exact, (if there does not exist any completely correct documentation of the process that is followed into the last detail) exploring the process along these natural units is the only possible method of getting information about it – one simply must 'go and see' (go' **n'CEA**) and talk to people[47].

Even if a whole process is supposed to not only be modified, but to be completely restructured, not understanding the natural surrounding and context, especially not the actual differences in the context for different sub-units (separated contexts) is not a very good starting point. Disregarding the details of a process will lead to decisions that are not based on correct management perception or well-thought-out-plans, but on

[47] Which, to remind the reader, is an integral part of the Toyota Production System, introduced earlier.

rough ideas that just might work or not. And even if it works, such behavior may put the actual burden of adapting and keeping a company running on the people running the process at the bottom.

Another point of view on the natural context separation is that the actual interaction within a process is defined by (maybe only temporally exiting) organizational structure – the actual "natural"/existing process organization. Meant here are the resulting informal structures if things like cross organizational teams and workflows exist. This process organization structure may often be completely different from the formal organization or organizational structure that the people actually involved formally belong to.

Since by definition all process modeling does focus on process organization, S-BPM brings both organizational concepts closer to each other, because parts of the model (the subjects) can also be mapped to personal/ bureaucratic structures (simply by labeling subjects with such terms as: member of department X).

Anyway, the baseline of the modeling complex scenarios can be stated as following: separate a structure to be understood/handled/controlled into manageable units (established in the previous section). S-BPM enables to do exactly that with the intuitively most logical, and natural unit in a business process: a human being and its personal process space. And for that S-BPM does not require the modeler to constantly 'translate'[48] between model and the real world either, but simply draw the model while exploring.

[48] 'Translation' means having to map back and forth between process-steps or process-order in the model, and sub-units aka existing persons in the real world. Something that can be done of course, especially by trained modelers, but it will cost capacity and get harder the larger the model gets.

1.5. Natural Context Exploration

The Idea of Natural Context and Exploration

The title of this thesis is ‘natural Context Exploration Approach’. What is meant here can best be explained with a metaphor.

It is based on the idea that a running business process is akin to a river running through a rain-forest ecosystem (the context), transporting and augmenting water (the result) from source to estuary.

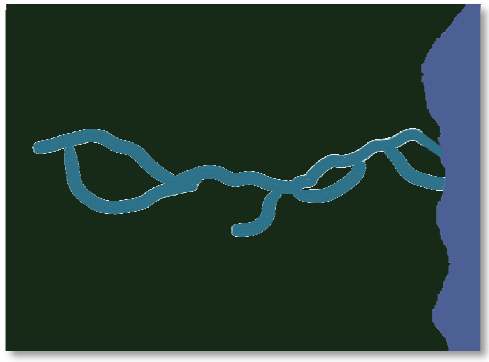


Figure 14: Interpreting a business process as a river through a jungle

The quality of the water that gets to the estuary may not be the best (polluted by ‘muri’, ‘mura’ & ‘muda – or basically too slow or quality of the process result too poor). But it gets there and the eco system that runs it is stable and has usually been for quite some time.

The quality of the water is important at the estuary, because there are the big cities that need the water (the customers that want their benefit fast and without mistakes). That is the point where the water quality is measured – directly via control or complaints, or indirectly via losing customers. And at some point the water quality will have deteriorated to a point where actions are required, that are aimed to change the process in order to improve the quality.

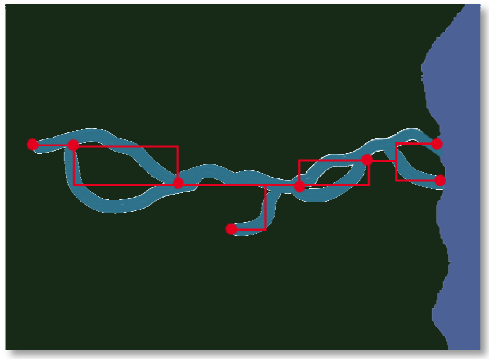


Figure 15: trying to map artificially structures onto the process from above

Satellite pictures from above (top-down process analysis) can give a good overview of what is going on, but also lead to the wrong assumption that the process can be streamlined or mechanically optimized. In real life the problems caused by such projects as the Aswan[49] or Three Gorges Dams or more general problems that come with human interference into existing ecosystem such as mono-agriculture especially in rain forest settings[50] show that even extensively planned measures can go very wrong or cause further problems than originally were meant to be fixed if the local ecosystem is not considered and instead measures or building projects are plowed through and natural flows are artificially channeled. This does not mean that it is impossible to do it, but it needs to be done carefully[51].

---

[49] In addition to displacing many families, the building of the Aswan Dam caused the Nile flood to cease and not fertilize the river banks as it had done for nearly 2000 years, reducing harvests and causing poverty.

[50] Among them are: the lowering of the water table, reduction of biodiversity, increased vulnerability to soil erosion, storms, diseases, pests etc.

[51] An interesting story about such a failed attempt to top-down transplant and reform local processes can be found in ‘Fordlandia’ [Gran09] . It recaps the story of famous Henry Ford’s

The ecosystem of a process is its context that has been build over time and makes it robust for many eventualities that may occur.[52] Mindlessly tearing down such grown, usually robust, structures might most often cause the whole system to fail. As with a real jungle, to understand and consider the local ecosystems through satellite pictures alone will only take you so far, especially if interpretation involves concepts and mindsets derived and built in other sectors. There may be small but important springs of the process river hidden underneath trees, or even in underground passages. Or there may be special plants or customs of the 'natives' (the process participants) that may stir the water and cause quality to drop, but prevent it to drop even further in special circumstances. All that may not be visible from above. To actually understand the jungle, and consequently plan reasonable measures or constructions to improve the process, an expedition along the process river is necessary to explore the context jungle (bottom-up approach involving the knowledge and participation of the process participants at the working level). Herein lies the actual difficulty to distinguish between necessary tasks and steps, and customs that merely produce waste with no predetermined measure to define good and bad, because it is context-dependent.

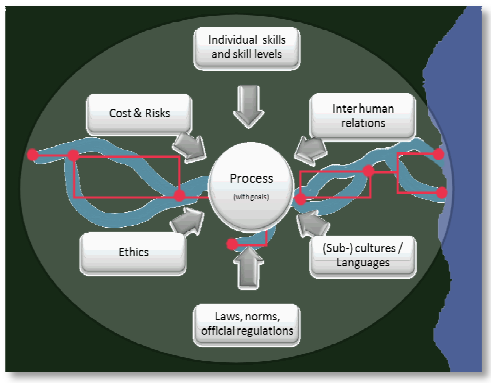


Figure 16: Process context in the metaphor

This context jungle, however, is no single entity spanning the whole range of the process. It is fragmented and may differ in many smaller sub-areas of the process jungle (theoretically for every single individual human, but at least for groups of process participants), that of-course may overlap and that are interconnected, but are not identical and come with their own needs or problems (comparable to the context separation possible with S-BPM).

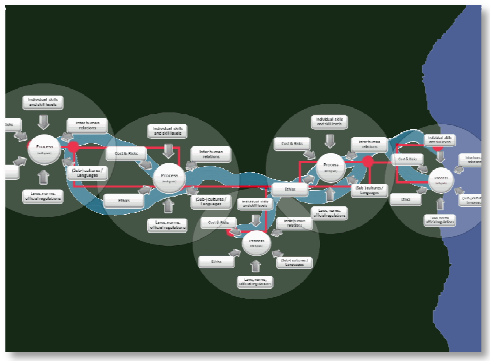


Figure 17: Actual Situation Multiple 'contexts' along the process / schematic of only partly overlapping context in a business process structure

At what point the expedition starts depends on what is possible. Pragmatic starting points are of course on either ends of the process, but only if these ends are clearly defined, and can be 'reached' by the expedition team easily (because the actual stake holders at those points are available). What is important, though, is that the expedition is no leisurely river cruise, scheduled within an exact amount of time, with pre-planned excursions or similar luxury, even though stations of the exploration journey may have been determined before from the satellite pictures (meaning upper management usually having a good general idea of what is going on and knowing to whom to turn and talk). If the exploration of certain areas (asking questions, understanding circumstances) gives indicators that at previous instances, things where overlooked, or

---

unsuccessful attempt to establish an American style industrial settlement in the jungles of the Amazonas in 1927.

[52] E.g. Because such eventualities have occurred needed to be fixed and have caused internal restructuring

interaction between ‘natives’ needs clarification, the expedition must be allowed to return to those places (the Deming Cycle concept and agile project management approaches).

Nevertheless, the overall goal of such an expedition is still the introduction of new concepts, technologies, or other changes into the jungle. In our fast living world, the construction teams are usually not far behind the expedition (developers of IT-infrastructure that actually, as it later shown, are working quasi in parallel to the exploration team). The idea here is that instead of simply razing something where they go, and planting some telegraphy mast in someone’s backyard, location and form of introduction is worked out together with the process participants and the exploration team, whose job is not only to draw maps (document the process), but at the same time educate the natives, make them comfortable with the changes to come, and even let them themselves find elegant ways to solve local problems[53] (*kaizen* and bottom-up principles).

Interim Wrap-up

Now this metaphorical approach tries to bind all concepts and ideas given prior into an intuitive synthesis that describes how to proceed when tasked with the job to develop and bring changes into an existing socio-organizational structure. One of the realizations that are central to the approach is that no matter how strong or ingenious any technology or process idea is, it is naught if it cannot be fitted into the environment it is meant for. This must not mean that the environment is not changed, but that if it is changed it should only be done incrementally.

The creation process of a model of business process is the central aspect of such a task, as it helps to put order to the complex circumstances of reality. The model can then function as the basis for future development which in turn can be demonstrated with the help of the model. But modeling needs to involve the persons affected (the process natives) in order to be able to really capture all conditions or contexts that surround a business process. Due to the core element of reflecting on or thinking about one’s own business process (one’s own work), the time of model creation can and should be used at the same time for continuous self improvement on the process details. If thoughts are already spent on analyzing the process to write it down, then why not just go a little further?

The structure and notation of the model can greatly help to support such a human-centric creation process as is described here. And with the S-BPM notation, a good example was given of a modeling notation that is ideal to support people in this task due to the natural structuring and separating of a business process that it implies.

Now the next section describes what actually will be done to use this concept in reality, including solving or avoiding problems that might arise while ‘exploring’

[53] While at the same time account for the overall situation or at least what has been discovered so far. This includes of course sometimes forcing out old habits, if no conclusion can be reached and local resistance exists only for the sake of resistance. “Trying out” things in order to get accustomed to new concept here is a good idea.

2. What to do?

After all the rather formal statements in section one, this section now will cover the pragmatic aspects of the approach.[54] The question now is: how does one actually explore a business process and its context, and what is a good structure for such a system development project?

nCEA basically proposes human interaction, communication and consideration of and with real human beings as the main focus for work, since the process participants are the ones with the actual knowledge about a current process, and at the same time they are the stakeholders the final product is meant for.

The overall motivation for this business process analysis is, of course, the premise that someone or something requires the process to be changed/to get better. A goal is what a "product owner" wants to achieve with the project. As laid down in section one, this should translate into gaining understanding of a (complex) business process and use the knowledge to create a model that will have consequences for the process.

2.1. Basic Proceedings

Even if unrealistic, in a linear order the steps of a basic proceeding would simply and roughly read like this.[55]

Eight Steps

1. Set a goal or objective and scope for the project; frame, boundaries and a general partition for the process to be modeled
2. Determine the parties/persons involved in the business process
3. Interview persons involved about who does what and why, and with whom they interact – elicit information
4. Use the information to create a correct model.
5. Check the model for its adequateness.
6. Use the model to create an IT System or tools to support the process, or derive ideas (maybe by doing a formal analysis or other methods) to improve the process through changes.
7. Implement and teach the required changes or new tools to the process participants.
8. Check if changes have had the intended impact and evaluate to what extent

A more detailed description for those points shall be given, together with thoughts on what may be problematic and why this linear order cannot and should not be followed in real life.

---

[54] Major parts of this section are based on personal experience as well as the teaching materials of Metasonic AG for their own BPM solution [Meta10] . Additionally as research for this thesis several expert interviews were conducted. A brief summary translation of statements given in each interview can be found at the end of this thesis.

[55] For now without defining who – project staff – does these steps, how and when, and expecting that no prior process analysis was done

1. *Set a goal or objective and scope for the project.* A frame, a limit, boundaries and a general partition for the process to be modeled: Actually this could be a quote: *"Step 1. Define Process Scope"* (in this case from [Silv09a] , but probably most other literature on process modeling). This can be considered basic knowledge. Every project needs a frame, a defined border, and a starting point. Otherwise there would be no end. This starting point and general scope is usually given by the nature of the modeling project itself, since someone must have initiated it with a purpose. Such a consideration is practical, but formally, it is against the otherwise proposed bottom-up approach in modeling, since setting a scope a-priori can only be done depending on existing ideas, concepts, and expectations about process and the project. As these might prove not to be accurate or adequate, this is a first step, necessary as a beginning and for the next steps, but one that should be changeable.

2. *Determine (at least one of) the parties/persons involved in the business process:* Depending on the given scope, it should be possible to identify people native to the process – the subjects or units in the process system. They are also stakeholders and at the same time the information carriers – in one person (a.k.a. end users of a possible IT system, a.k.a. domain experts). Identification of those persons might be via the detour of functional roles or departments, but in the end there should be a list of actual real people with knowledge that will cover the process. This first circle of persons involved in the project will or might be expanded if discovery during the project turns up further people with possibly more or other contradicting detailed knowledge about certain issues.[56] As an example: even in a small and relatively simple trial project with a research customer initially involving no more than four persons during discussion, it turned out that a fifth person would know certain aspects better and thus was summoned and involved.

3. *Interview persons involved about who does what and why and when, and with whom they interact – elicit information:* Or "go and see".[57] But easier said than done. The simple idea is this: you meet with people, ask them who does what in the process and why, especially with whom they interact and when, and record that. Usually, this cannot be done within one session (lasting one to two hours) for several reasons:
   Especially in initial projects, people (especially in service sector) are not used to "process-thinking" or reflecting on their own work.
   People have to be convinced/have to learn about the purpose and methods.
   The actual effort required for process analysis is often underestimated in the beginning. Especially if it turns out that the task of a single person or area is more complex and takes more time to be recorded and understood.
   People get tired and/or need time to reflect on their work, so different information might turn up at a later time
   Information turned up in an earlier session might be conflicting or inconsistent with information gathered later from a different source and may need

[56] Extreme cases would be that either only a single person or already all persons involved included there (the last one being rather unlikely)

[57] (and learn for yourself what people are doing) – this is still a very good approach and an essential part of the Toyota production Way for a reason. See [Like04]

clarification. Especially naming conventions and terming issues may arise sooner or later.
Information given will depend on the surrounding (single interviews or group discussions, presence of audience like managers, coworkers or no). There is no clear “best” solution for this matter. Best advice is to go with what the person involved claim to be most comfortable with and try out what works best.

4. *Use the information to create a correct model:* Naturally, this is the core job and should be fairly easy given a modeler versed in the notation used. But as has been stated in part one of this thesis, models do grow over time. Modeling itself can be rather quick, but during modeling, simple mistakes can happen or further questions about the process may come up due to the need to actually think about details. Furthermore two ways of modeling may be valid but have different consequences for a later incorporation of the model into an IT system. In such a case, it is a question of adequateness of the model and as such may need to be answered e.g. by end users. Due to the need to validate a model this step is usually closely interwoven with the previous one, going as far as the possibility of modeling while interviewing stake holders. This should be done if project staff time schedule, modeling notation and tools allow this kind of real-time-modeling. And even if not in real time, it is a good idea to simply show and explain the model to the process natives from time to time to verify correctness of the statements as well as show the participant what was created with their knowledge. Most modeling notation should make it possible to present models to laymen if they are being guided and the model is explained carefully.

5. *Check the model for its adequateness:* This step actually cannot really be done at this point, because it is the next step that describes adopting the model for its purpose. Only after that can the actual result and the expectations be checked against each other, and thus it can be verified that the model was adequate or not – or at least adequate concerning certain issues (steps 6-8). It would simply be great to already have an adequate model before starting with actually developing something.

6. *Use the model to create an IT System or tools to support the process, or derive ideas (maybe by doing a formal analysis or by using other methods) to improve the process through change*: In other words adopting the model for its purpose. The actual work and most important decisions are contained in this point, and dealt within a rather short description here. The nature is vastly dependent on the chosen modeling language and tools and because of that it is impossible to go into detail. Whilst with solutions like Metasonic BPM suite – where the model actually IS the core of the IT system and development is limited to connecting the core to additional tools – allow changing the model whilst developing the system, other approaches and tools may need a completed model as a working basis and allowing no more changes afterwards. There are tools and projects trying to automatically create source code or stubs out of UML diagrams for example.
   In any way, the typical problems while developing IT-*System* will apply here: estimating efforts and cost in advance, handling the risks involved with them additional wishes and requirements will pop up, integration problems of legacy tools/systems, and that is naming only a few. Requirements (or user stories)

should have been captured during modeling process and should actually be in the model.
It should also be noted that is vastly unrealistic to wait with development, until a model has reached a certain maturity.

7. *Implement and teach the required changes to the process participants (train):* It might be the case that there are indeed no changes to a business process after all the previous steps, but this is rather unlikely. As such, the intended changes need to be implemented into the process framework; usually with consequences. But change management is a form of art itself. Following the gathering of information, changes ideally should be at least partly considering the context. Impacts and consequences should be at least estimated before trying out new ideas, making implementing them easier. Still people by nature resent sudden changes and thus need time to get used to them. At least the participants may need to be taught or instructed into new techniques, tools or workflows. Involving the afflicted persons from the beginning in the project is the idea here. For example by explaining possible influences and the likes of the modeling project and asking them for own suggestions or improvements during interviewing modeling or verification sessions.

8. *Check if changes have had the intended impact and evaluate to what extent:* Based on the idea of [Dörn09],[58] one of the fundamental mistakes made in working with complex systems is not to check whether a change or other measure had the desired effects. So it is imperative to try to figure this out, or to measure how strong and in what directions effects took place. How or what to measure should have been determined before, e.g. during an initial session or depending on problems discovered during interviews. In the worst case, an unfavorable evaluation could mean that the whole development project needs to start anew. A complete failure though is an unlikely outcome, but further changes to scope, system, or model based on the evaluation are possible.

Iterations

The previous section has already indicated that there can be no clear separation between these stages. Many things that can (and should) happen parallel to each other, so synergy effects can be used, many might have influence on each other, and some steps might not even be necessary under certain circumstances.

E.g. a lot of process modeling and planning might already be done in the first few meetings, because project teams on customer side have already prepared material. Or interviewing might not be as necessary because process analysis and modeling was done before in a different notation and can be used as a quick introduction for external personal.

Figure 18 depicts a graphical representation of how these steps could be interwoven. It should be obvious that there is no real order or schema at least if one was trying to

[58] See the paragraph “Six Errors to be made during the Handling of Complex Systems” in section 1.3

account for all possible input. The other interpretation is: there is simply no way to plan ahead or formulate a working plan a-priori.

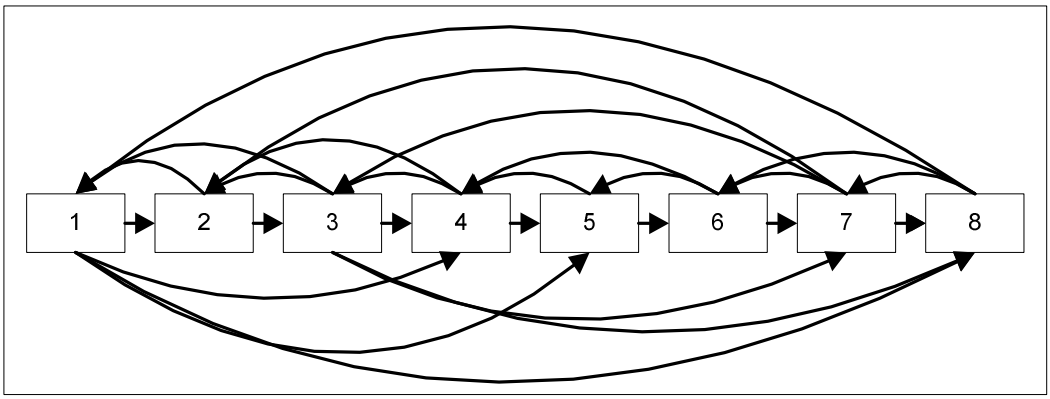


Figure 18: visual representation as of how steps of the basic proceeding could be interwoven

[Hürl09] , describing steps in the model construction process of Vester's Sensitivity Model [Vest02] , finds the following words that can easily be applied here as well, even though model generation is only a part of it:

> ***Theses nine steps are again not just sequential, but iterative in the sense that results of any step can yield insights that lead to revisions in any earlier step.***

The same basic problem, in that case on the later stages of nCEA, the system development, can be found and is addressed in pretty much any literature on agile software development (most prominently Extreme Programming and Scrum in e.g. [Beck06] and [Cohn10] or [Larm09]). The answer given to solve this problem is simple: live with it, embrace the continuous changes, be pragmatic, and focus on getting things done to produce and outcome. Allow for dynamic, agile iterations, by using time-boxed intervals with repeating "ceremonies" and procedures as fixed frame, to decide and plan concrete actions[59] primarily for the current interval.

As stated in the section about agile development methods SCRUM in particular has been established as the methodology or frame work of choice for agile development projects in many domains. Because of its simplicity and flexibility it was used as for this thesis will describe how to adopt SCRUM as a frame work for explorative projects concerned with modeling and implementing business process.

Part two will continue here with discussing in more detail several pragmatic aspects and techniques that are of importance in the execution of

2.2. Thoughts, Concept, Tools and Techniques for Execution

The ideas given in this section do not follow any particular order. Nevertheless, they should be kept in mind when executing an nCEA inspired project. It should function as a kind of expedition hand-book with some general tips and ideas to keep in mind.

[59] Actions like concrete interviews, review or coordination meetings, validation session, or in

Validation of a Model

Creating a model is a process done by humans. As such errors – e.g. leaving out aspects, forgetting to complete notation or misinterpretation of 'signs'[60] – will happen during designing, and consequently must be found and corrected.

Validating a model has three aspects.

The first aspect is the correctness of the model itself in regards to its notation. With modern computer-based tools this is usually by far the easiest and fastest validation process, as it is usually done via the click of a button that will let a computer routine check the model and inform the modeler about the missing connections or definitions if they are syntactically necessary.

The second aspect of validating is harder: making sure that the 'signs' of the model are not contradicting information gathered, that they are not leaving out parts or important details, or at least that they do that on purpose. Model review by other humans is the key to that, as no machine can know about reality[61]. This can only be done by people with indebt knowledge about the process itself – the process participants. Depending on their comprehension of the modeling notation this can either be done simply by sending the model (or part of the model) to the respective persons, or the model needs to be presented in person including a detailed description and explanations. S-BPM models have an advantage for this task, because they are already person-centered and have a rather easy notation, making it quick and simple for non-modeling-minded persons to understand and check their own workflow without need to understand other parts of the model or the process that might come in between.
Also helpful are tools that allow for running through the process and demonstrating the steps with a "validating" audience without concerning them with the model itself.
The overall goal though stays the same: ask and figure out if all possible/needed process paths are covered at what points.

The third aspect of model validation though is the most complicated: how to make sure the model is adequate? This can be only decided by the people having to live with the consequences of the model (e.g. in form of a workflow engine – after its commission). To minimize major changes resulting from lack of adequateness after the whole model/system has reached a certain level of maturity[62], it is necessary to try to estimate this factor from the beginning. Prerequisite for that is that stake holders can learn and estimate consequences for them and voice possible concerns[63] as soon as possible. Making communicating the according information necessary for any meeting between

Systems or approaches allowing rapid prototyping or systems like Metasonic BPM suit, that allow for demonstrating consequences of and correlation between model and IT system with a relative ease, are vastly helpful, because they allow a wider circle of person to understand or directly see consequences of designs even if they do not have profound knowledge about the IT system itself.

---

[60] Signs in the semiotic sense, that the items drawn in a model do represent something in reality. How a sign is read though depends on the reader.
[61] Which is usually the point of the model anyway: teach a machine about reality
[62] Major changes at later stages of development usually tend to be harder and more complicated to execute and as such are usually more expensive in a business context.
[63] As always this does not imply the need to try and account for every concern or doubt – which is impossible. But hearing people out and explaining is usually helpful in any event.

The Model as a Tool

Within nCEA the model – meant here is the actual visible and digitally created diagrams and charts – is considered not only the central product or artifact to be produced, but also a central tool by itself. Visual representation of complex matter is always a valuable help for humans - for communicating among each other as well as with process participants.

In the end a process is still something abstract, an idea about a continuous development. One cannot touch or feel a process directly, and humans often perceive their own task not as part of a larger process, but only part of their own personal work process – their rarely think in system categories.

A model can give substance to a process, as an object for all participants to see and to discuss the process – which is in the end only the idea of several connected task and decisions. This way it also serves as a tool for synchronizing what is in the head of people, so everyone actually does talk about the same thing.

For [Larm09] this is even the most important aspect with model creation[64]:

> ***The primary value in diagrams is in the discussion while diagramming – we model to have a conversation.***

For this to happen of course the creation process must allow for such conversations between (the right) people.

As a final thought at this point/another thought: A Simple modeling notation – making the whole modeling system easy to understand for all participants – is of especial value for the sake of communications. Faster understanding of what is actually depicted in a model – and understanding the consequences – is a big advantage (See 'Advantages of S-BPM Models' section).

Thoughts on Communication

> ***"It is commonly assumed that as long as speakers share a grammar and vocabulary, they can always make sense of each other's statement. The listener's interpretation of the sender's message, though, may be entirely different from what was originally meant by the sender. They may be exchanging words, but not necessarily meaning." from [Nou98]***

Inter-human communication is key and central aspect in exploring business context. At the same time it is a major if not the biggest source of errors and mistakes, and problems in general. Exploring a business process always involves communication and as such the explorers should at least be aware of the possible problem causes.

One of most frequently cited explanatory models for communications is the 'four sides model' of Schulz von Thun [Sch08]

[64] Though for Larman is talking about a development process where the model is not an important artifact or actual product

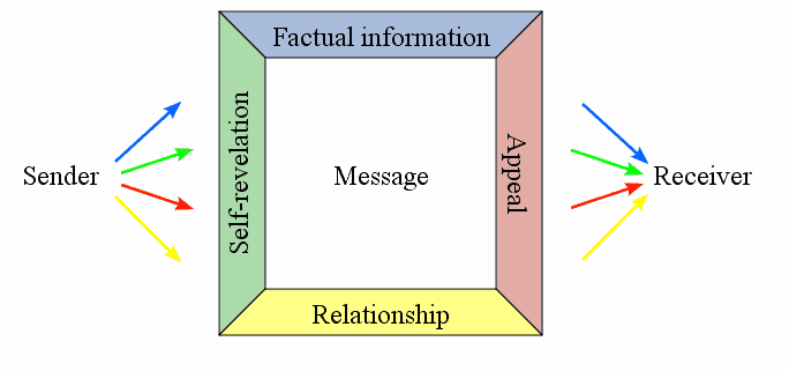


Figure 19: Four sides communication model of [Sch08]

The idea of that model is that in human communication there are four 'sides' to every message. Each of these sides depicted in Figure 19 is akin to channel transporting certain information about a certain aspect.

The *'factual information'* or '*matter layers'* does convey fact-like data and facts itself. It contains the actual message in terms of technological information transfer.

The other three sides are often not that directly present, but interpretation on the receiver side will usually try to figure them out.

The '*self-revelation'* side contains information about the speaker – his motives or values. This can for example be a certain emphasis or stressing of words to indirectly express like or dislike of concepts.

In the '*relationship'* layer certain aspects that express the standing or relationship between sender and receiver. In certain languages like Japanese there are actually formal grammatical structures to explicitly express this side of a message, but also in European languages form, or choice of words are usually crafted in ways that are dependent on whether one talks to a (perceived) superior, equal, or lower-ranked person.

Finally the '*appeal'* may directly or indirectly convey commands or wishes, even if the grammatical structure of a message does not contain it. E.g. saying "It would be better if you leave now." – is formally just a statement, but obviously does contain the appeal to actually improve the situation.

All these 'sides' are always present in any communication and on each channel there can be mistakes – even if the message is completely received. Mistakes can happen either on side of the sender, who is not 'coding'[65] his message right, or on side of the receiver who can misinterpreted any of the information transmitted on the four sides. Additionally the chance of miss-communication or misunderstanding is highly depending on the context and situation of the communication.

[65] Coding by e.g. formulating, choosing words, intensity, pronunciation, or dialect in the act of speaking or writing.

Tactfulness and sensitivity towards the own choice of words and the good-will to interpret statements in favorable, forgiving, and cooperative way is the baseline mandatory for successful projects.[66]

An additionally layer of problems does arise in communication when there is actually no or only a partial share of grammar and especially a vocabulary, between sender and receiver of a message. More concretely if e.g. the conversing people have different mother-tongues.

> ***"Cross-cultural communication including that between Japanese and English language speakers is, in a word, complex."***
>
> ***– quoting [Nou98] P. 82 again to convey the warning.***

But cross-cultural is not limited to actual ethnic-based cultural differences. Cultural differences already can be found between two departments of a company that may have very differing customs and especially vocabulary. The simplest case may be that were for the same object one department may use a different term than the other (e.g. what is send as an 'order' is processed as a 'request').

Aligning and before that actually discovering the 'cultural' differences inherent in the interaction of the participants (and their personal process spaces) along a business process will be part of any exploration, and thus should be expected and prepared for.

In actual work this will translate into work-task involving discussing and agreeing choice of terms with the process-natives, and also maybe keeping a vocabulary list to track according agreements. Also sometime the use of experienced translators (between foreign languages as well as department slang) may be strongly advised in order to progress with relative ease and not get lost in misunderstandings. As with the interpretation of natural languages though only experience really does improve such performance, and as such learning to speak a common language must and will be part of the joint- learning teaching process that should occur during the project.

Furthermore the state effects here are of importance for a whole project and should be considered when introducing it to the people that are going to be involved. Misinterpreting or misunderstanding the goals or ideas might hamper the whole project.

The intention of this section was to mention communication problems that might arise during the project of modeling and improving a business process, while interviewing process natives to get information. But such problems might be actually an important part of the general problem within a business process[67] itself. Solving or at least managing inter-human problems can also be part of the improvement process itself. This might include such things as changing interaction workflows in a business process from indirect to direct real-time communications (or vice versa) – or changing the

[66] Helpful literature based on the four-sides-model can be found e.g. in [Sch04] or more group oriented in e.g. [Sch03]

[67] The model of Schulz von Thun is one example of how personal process spaces in a business process can be connected in more than one official way. In this case there are four channels instead of just one that might be expected when two people exchange messages.

initiator of a communication process. E.g. making an active phone call instead of waiting for a written application.

Interview Guidance

Interviewing people could be a whole chapter or book on its own, and in fact there is tons of literature in existence one expert interviews and problems and challenges this kind of qualitative research brings - like: [Bogn09] or [KvBr09] to list an English language example. Additionally there are many side factors as interpersonal relations and experience connected to this topic. And despite all theory the success of interviews depends a great deal on the personal experience and people skills of the persons conducting the interview, as well as the context the interview takes place.

Because it would go far beyond the scope of this thesis this subject will be limited to a few pragmatic advises:

The first is: **Record and Review interviews**. Not to collect evidence and nail people down on the precise formulation of statements they have given, but simply to be able to review the discussion or interview afterwards. It is nearly impossible to keep all information in mind or directly input them into a model. This is especially true in interview-only sessions. The cause for this is our limited attention span at every single point in time. A human being can (usually) not process all given information at once. E.g. Limited attention span, maximum of 7+-2 items on mind at one time[68].

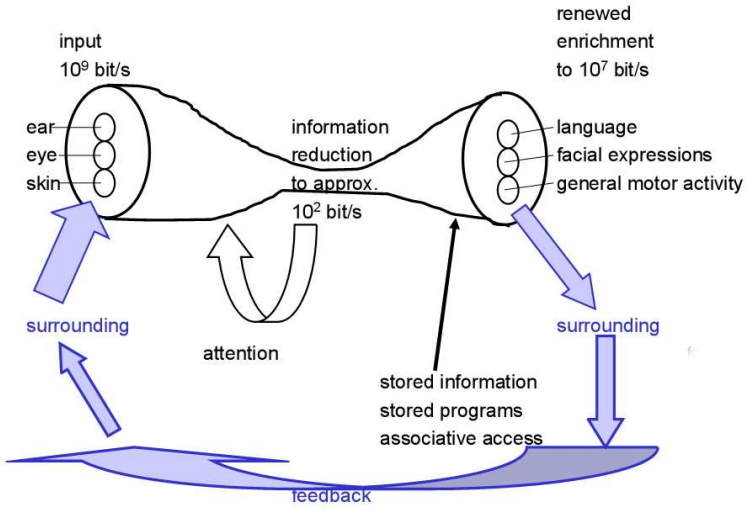


Figure 20: Bottleneck attention (Following [Vest02] p. 23)

Recording and reviewing is therefore a most pragmatic advise to overcome that kind of weakness in human perception and avoid the according chances for errors.

Participation: (mostly) voluntary participation is an advantage. Interviews should not be akin to police grilling a murder suspect or a fight for information. Ideally they should be conducted without pressure of any kind. It is unlikely though that every person needed to be interviewed for process information will do it because he has nothing better to do.

---

[68] The phenomenon of the human relationship to the number 7 was first publicized by George Miller in 1956 with his article "The magical number seven, plus or minus two: Some limits on our capacity for processing information" [Mill56] .

Do it on site: there is no use in getting people out of their work environment (and usually it is expensive, too). Even though modern telecommunication allows for remote interviewing and discussion, still nothing beats being there at least for a first impression and grasp of the involved persons.

Questions: the basic questions to be answered are simple: what, why and how[69] do you do your part in the process and especially with whom do you communicated when, and what is the structure and nature of the messages. All this is one of the points were an explorative method is necessary (or dialogue) as usually no person can answer the questions in regards to all process steps in the first moment – especially definition of messages – and thus needs time to think about own work .

Try to uncover the not-so-happy scenarios: important for business supporting IT tools are usually that they at least try to cover special cases or exception. One of the main reasons to involve people deep in the process is not because they know what to do in normal occasions – the ideal process path is usually known to the "higher ups" – but because they usually have an idea what can go wrong and how to solve problems depending on experiences in the past.

Explore the root cause: Base on Toyotas improvement philosophy: "*Five times ask Why?"* [Toy06] Or better direct people to the point, where they ask why and try to give the according answers. Sometime it will be necessary to not trust or be satisfied with the first answer and repeat question in order to get there. Also trying to understand why certain thing happen is of utmost importance before setting changes in motion, especially if not the right root cause of a problem has been identified.

Interview with a manager (product owner) present: whether this is a good or a bad idea depends on the manager and what his expectations and relationship with his people. A Manger can or should functioning as a kind of guide and translator and can help get process information in the correct order and faster. At the same time he can use the process analysis session to coordinate or agree on new rules or adjusting the workflows of his personal, integrating them directly in the process model.

Group discussions: larger groups (e.g. around 15 persons) tend to be impractical especially if everyone only has to be interviewed for his own part of a rather linear process. The guy in the back will get bored whiles people before him tell their stories. In such cases it is better to do follow up interviews one after another. More interesting and important are smaller groups of people that actually interact during a process, do communicate or exchange information otherwise. It seems to be a rare occurrence that people – especially of different departments – agree from the start on how to describe or model their communication and interaction. Letting the directly involved people solve the problem (by guiding their discussion or making suggestions) is usually more efficient and faster than trying to figure out a solution for contradiction alone. If not discussed, solutions not agreed on by all people involved will often lead to conflict. The importance of issues to be agreed on though is again subject to

[69] The first question of the 'Five-Ws' 'who' has at that point already been answered – you are talking to him at that moment in most cases. Though if talking about work of other people that is still the most important question. In that case it should also be distinguished between 'must' be done by another person and 'can' be done.

Follow up interviews or repeated sessions: sometimes it will be clear within after one interview because everything has been said, but often time in between and other information input will lead to the need for follow-up sessions. Single person interviews should not last more than 1-2 hours, and larger time in-between sessions (e.g. a day) will give time to reflect on said and learned things, try to model facts

Pairing or Teaming Up.

Pair programming is a (not undisputed) cornerstone of agile development (especially Extreme Programming [Beck06]).

The base idea for it is that two people share (for a time) a workspace with one desk and one computer. While one is typing the other is supposed to check for errors. Both should continuously share their thoughts and goal to their actions. Of course:

"The collaborative activity of pairing is dominated by communication: […] Understanding is shared and affirmed") From [RoSh10] p. 5. – Thus, of course, the general concerns of the 'Thoughts on Communication' are present here all the time.

There have been several studies on the effects of pair programming such as [LuChNo08] , [ArGDS07] , or [CoSiSu09] .

The effectiveness of pair programming is – as mentioned – disputed, and scientific results are mixed. The main disadvantage lies within the fact that two fully-paid developers do only one thing at a time which in linear logic translates at a first glance into double the cost for the same work or at least slower overall work progression. Also often noted as a disadvantage is the fact that sometime people need time on their own to think things through or to get their head clean. Constantly sitting together is likely to cause stress.

On the upside if applied there is a significant reduction errors while the actual loss of work performance smaller than the to-be-expected 50%. Additionally a number of positive factors that are harder to measure are often listed when proposing pair-programming. Among them is better information distribution within Development Teams, better problem solving ability, or better design quality (due to constant peer review). It is also most beneficial for training inexperienced developers and improving their working performance.

But "*you cannot expect faster and better and cheaper* [at the same, ed.]" as a meta-analysis of several studies on pair-programming reminds, that also "*suggests that pair programming is not uniformly beneficial or effective*" [HaDAS09] . Meaning is it highly dependent on the situation it is applied.

The baseline as it seems to be: do it regularly, but not necessarily always. Review in pairs the work done alone – test it, describe it, correct mistakes. But never force it upon people.

For the modeling side[70] of nCEA there are several possible scenarios to use such pairing or also small group work of three or four people teaming up.

---

[70] In contrast to the coding or IT-System development side, where the normal technique and of pair-programming applies.

Most commonly will be the idea that a modeler and a process-native are sitting together, analyze the process and model (or validate a model) at the same time. This set-up allows for great insight on both sides[71] and should be practiced if possible. But it also takes time – longer than interviewing and solving modeling issues afterwards. Even if the according process-natives are available at all, their time might often be limited. Also at least on interviewed process modeler has directly state that he prefers modeling 'at home', because even expert modelers need time to figure out certain issues or find ideas how to express or model certain circumstances.

The other scenario for pairing up is that of two (or maybe three) modeling expert working together and try to bring form to a business process model – based on, or while reviewing information or interviews from . This is most akin to the classical idea of pair programming. Personal experience has shown that especially for novice modelers this procedure is beneficial in both terms: learning the modeling notation and also post-analyzing information of process-natives. But also here pairing should not be done continuous. Instead regular intervals to synchronize ideas and work done in between should be the course of choice – if there is more than one person involved with the actual modeling at all.

Keeping and extending Ideas.

No modeling notation will probably be able to hold every useful bit of information about the complex system of business process. Through comments to certain parts of a model it is possible to capture a wide range of additional input information that might be useful transferred to the right people. This is especially important to support the *kaizen*-spirit.

People should be encouraged come up with and their ideas for problem solving or improvements. Important for that is, that such ideas reach the right ears or are discussed. Furthermore feedback on the ideas is important, be it positive or an explanation why it will not work.

During an nCEA project it is most likely that people will have ideas or at least superficial thoughts that may only come up in one-on-one sessions. If an idea seems to be at least reasonable it should be recorded or written down, if it cannot be immediately discussed or if a tryout, test run, or required permission or approval needs pre-planning.

The idea is to have a kind of note card or other tracking down the most basic feature of the ideas. This idea-characteristics-note card might look like as following.

[71] Including maybe direct insight for process-natives to other parts of the process and the model including the possibility for direct question and answers.

Idea or concern: Originator: Involved subjects or persons (whom it may concern): To be discussed and implemented (date):

In the course of the project the collected ideas should be reviewed, discusses and where applicable tried out. After review or try-out according feedback should be passed back to the originator.

Keeping track of the originator (may also be a department or group) and giving feedback on ideas can encourage voicing of ideas since otherwise people may simply think it may not be worth the effort. Also giving people signs that their concerns have been at least acknowledge will help improve acceptance.

Since written note cards may not be practical where people involved are divided far in between or simply many people are involve, it might be a good consideration to have a kind of virtual idea-tracker on platform/collaboration tool or project management software that may be used.

2.3. Introduction to Measuring

Measuring the performance of business processes (especially in the service sector) is a controversial topic.

On one hand there is the wish and the need for hard factual numbers, derived scientifically. A need for tools to support decision, and that go beyond simple ‘gut feelings’.

On the other hand there is the problem that especially in with human interaction actually finding things to measure is sometimes impossible.

Additionally there might be legal problems. At least in Germany it is forbidden to use pretty much any measurement of individual work performance for evaluation by superiors. As a consequence it may be hard to even conduct them or get the support process-natives whose work is to be analyzed. Because once numbers or reports are finished the chance and the according fear that they find their way into the wrong hands with unfavorable consequences for the measured people is not always unjustified.

Consequently openness and trust is mandatory when measuring performance of people – especially concerning the handling and usage of result.

What can be measured

The overall goal is to identify units or processors in a process that causes problems – their performance should be measures. The units of business process of course are in many cases individuals or groups of individuals.

Among many other, maybe very sophisticated methods, there are three things that can be measured[72] to evaluate a processor: throughput over set intervals (of objects to be processed), (variance) lead times, and error frequency (number of errors made during processing as a measurement of quality).

Throughput is rather easy to understand: the more e.g. applications your organization can process the better it is.

Variance of lead times is trickier: in manufacturing there are automated production streets that work with one set speed, and thus without variance of lead times. In service industries this will often not be the case, with process task having rather large variance in lead-time. According to [Buch09] minimizing variance is the key to better running processes in the service industry. Alternatively process task with a high variance are the points in a process were improvements, e.g. in form of more workers, may be necessary.

The error frequency (if it is possible to measure errors) is an indicator were possible problems are, but also a quality measure for a whole process. A high throughput (due to tight production) is naught if 50% of the processed objects are erroneous.

Overall, especially with the error frequency measure, it is important to note that any measurement is neutral in the first place. In congruence with the previous section it is always important to make sure what the cause of a certain number is (five times ask why) and how it needs to be interpreted in the context of the process.

The methods to measure with can be widely varied. Usually base on evaluation of existing process records, but can also simply be done with stop watches or self-assessment by the process natives.

[72] Based on [Buch09] and [Proc10]

3. Adopting the SCRUM framework for nCEA

The problem challenged by this section is how to organize or orchestrate all the aspects, concepts and ideas discussed in the previous sections in reality. As it has been argued in section 2, because pre-planning in great details is in most situations impossible on the long run, an agile method that can adapt to and in corporate unforeseen circumstances is necessary. As stated before, Scrum will be adopted for that purpose.

The 'product' to be produced or developed under nCEA is an existing business process. The most important aspect of this 'product' is the creation and the use or implementation of a model of said businesses process. The model is expected to function as the basis for a supporting IT system. The process of getting to an adequate model of a business process is of importance, as is a prepared environment where the changes brought by the model can be implemented without causing large disturbances – to fit naturally so to speak. The understanding gained, the knowledge, and the training of process participants (natives), as well as new rules, decisions or changes in workflows concerning the process, are also part of the "product" and are not simply by-products.

For full effectiveness of this approach it is ideally to be used together with a model-centric information system development environment for business process support[73]. It requires the option of quick integration or translation of a model into an IT system, which can allow for faster implementation cycles and extends the range of the whole project (meaning not only modeling, but spanning from starting analysis to full system integration, allowing direct feedback on that). If system-development cycles (from model to system) take too long (more than two to four weeks) a normal project should be started with classic development teams and considerations. The environment should allow for relatively easy changes in the IT-system based on later changes in the model.

3.1. What Scrum is and its Components

Purpose of Scrum

Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques. The role of Scrum is to surface the relative efficacy of your development practices so that you can improve upon them while providing a framework within which complex products can be developed.

[ScSu10])

[73] nCEA was first created with Metasonic Business Suit and according advantages and possibilities in mind.

"Everyone will like Scrum; it is what we already do when our back is against the wall"

from Jim Coplin in [ScSu10]

Scrum is an iterative, incremental framework for project management and agile software development.

Although Scrum was intended for management of software development projects, it can be used to run software maintenance teams, or as a general project/program management approach.

(From [Wik10])

---

[74]The basic idea for Scrum development is that within fixed intervals of times – the Scrum Sprints – work is done on the project by a self-organized team – based on goals and priorities set by the 'Product Owner'.

Team size is roughly seven people. Projects of larger sizes are done by several team working in parallel. One of the team members has the title of Scrum Master and functions as a kind of ceremonial master to keep the Scrum process running. Finally there is a quasi-member on the team who is a representative for the stakeholder in the project, the Product Owner, who does no development himself.

The requirements of the product to be developed, that are required to finish the project successfully, are listed in the Product Backlog. They can be edited, modified or changed during the project. All items in the Product Backlog have a priority that is determined by the Product Owner based on information he receives from the team and the goals of the stakeholders.

According to these priorities some of these requirements are chosen, broken further down, and translated into doable tasks, and then taken into the Sprint Backlog. This is done during a Sprint Planning Meeting. The Sprint Backlog is the to-do list for the team for the duration of one sprint, a time box that usually lasts four weeks.

During a sprint the team is free to do (and complete) any of the tasks in the Sprint Backlog. The division of work is coordinated among the team members. A daily meeting, the Daily Scrum, is used to keep every team member informed about the work of his colleagues.

---

[74] Information in this section is taken from: [ScSu10] , [Wik10] ,

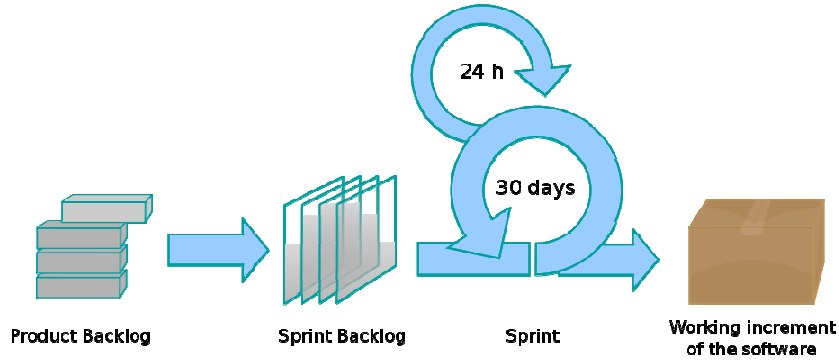


Figure 21: General Scrum process (based on [Schw04] taken from [Wik10]

The general goal for the team is to have a (at least partly) usable piece of product at the end of each sprint.[75] The product will thus be incrementally improved over the course of several sprints.

The progress of work done during Sprint and in general is tracked with so called Burn-down charts (for the general release and sprints). These are charts with time on the x-axis and work-left on the y-axis to track remaining tasks and estimate remaining efforts.

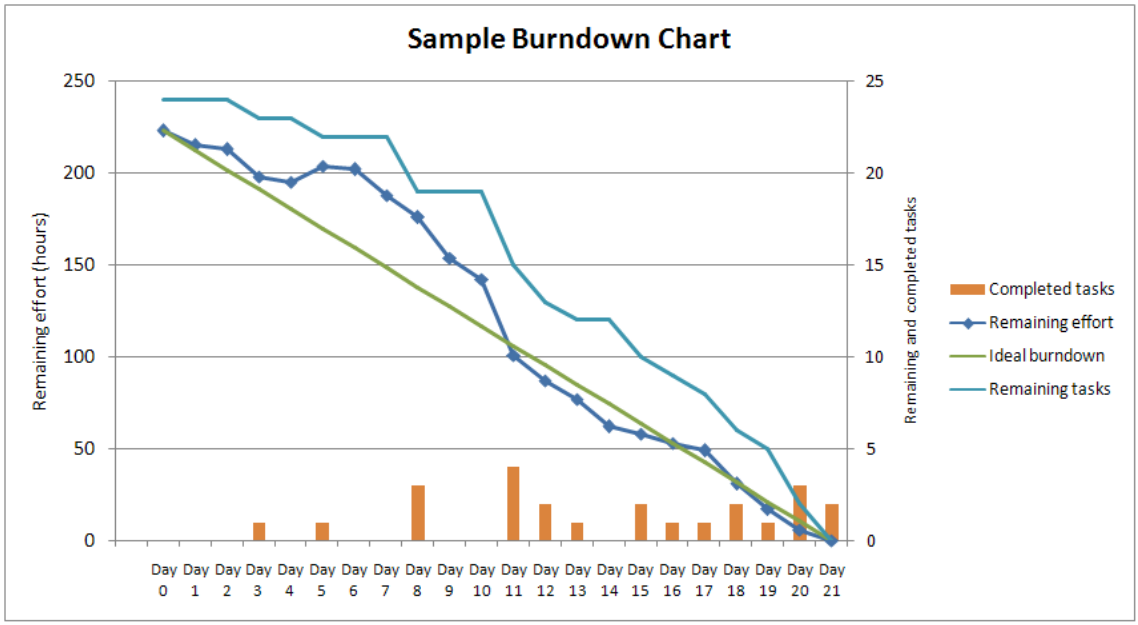


Figure 22: Sample (Sprint) Burndown Chart. (c) Pablo Straub under public domain license

At the end of the sprint there will be a Sprint Review Meeting, and a Sprint Retrospective Meeting followed by a new Sprint Planning Meeting for the next Sprint.

During the Sprint Review Meeting, the current product is presented to stakeholders and evaluated. Also work that was not completed is reviewed.

Finally during Sprint Retrospective Meeting the development process during the last sprint itself is evaluated and ideas on how to improve the work of the Process Team are discussed.

---

[75] This will not be possible or useful in every situation.

Figure 23 gives an overview over the cyclic structure within a Scrum Sprint

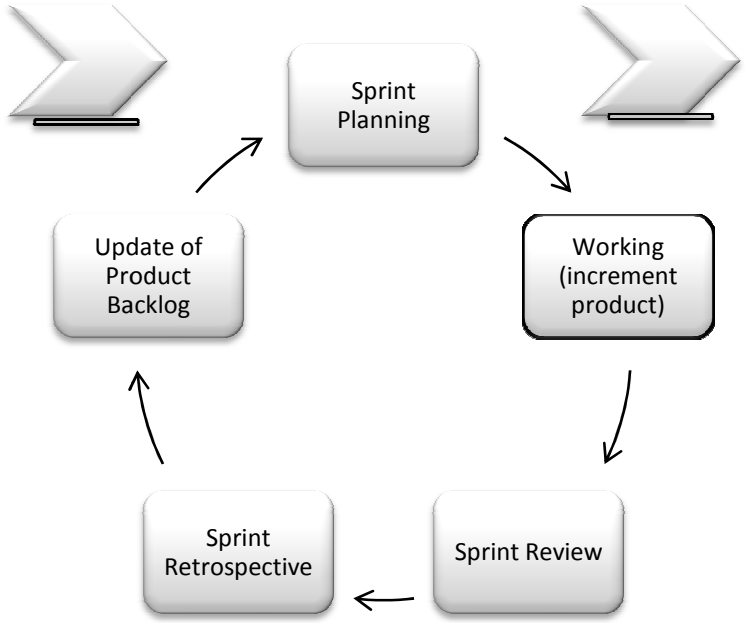


Figure 23: Conceptual cycle diagram of a Scrum Sprint

3.2. An nCEA Project

For nCEA a few changes will be applied.

Project Teams

Like all Scrum projects, nCEA projects are worked on by people organized in teams. In contrast to normal Scrum project the smallest unit in the approach is not one team but a joint-venture of two teams.[76]

A Process Team (or Modeling Team) – concerned with developing the process and its model, and a System Development Team, responsible for "using" or implementing the model and other information to create and integrate a corresponding IT-System or application.

Experience from reality[77] shows that a reasonable size to be expected for the development team is rather small (around three to five people). Process team in contrast may be larger (ranging from five to fifteen), depending on the process. In general the team sizes vary depending on the situation, too, and there probably will be cases where more than one team or group of teams may be required.

[76] For reasons of simplicity it will be assumed that there are exactly two teams. When in reality this might differ.
[77] See Expert interviews in Appendix A

Development Team

The work of the Development Team is rather ordinary in the sense of classical Scrum development with a Scrum Master, a Product Owner, and Scrum techniques, as Product Backlog or Daily Scrums.

The tasks on their Product Backlog depend on the work of the Process Team, though. So in contrast to normal Scrum projects, tasks may only arise slowly, as the Process Team learns about their process and comes up with detailed needs and requirements during modeling. Especially in the beginning, the Product Backlog of the Development Team may be rather empty and Sprint Backlogs can be limited to preparatory work, like setting up technical infrastructure and similar tasks (depending on the technologies chosen). Requirements or items on the Product Backlog may arise only slowly, and may sometimes even contradict each other or make previous work obsolete[78].

If the team finds this to be not working properly, it might be a good idea to try and use other Scrum modifications to solve the problem, specifically Scrum-ban [Knib09] that allows for more need driven and even less pre-planned working style.

Because the Development Team is otherwise assumed to be a normal Scrum team, all further mentioning of team activities or modification, and other uses of Scrum procedures refer to the Process Team, if not explicitly including or mentioning the Development Team otherwise.

The Process Team:

The idea for the Process Team is a little bit different from normal Scrum teams or their procedures. Thus there will be a few distinctions.

The team should consist of the persons involved in the process (the process natives) – meaning: a team mainly on the classic customer side, or a team made up of stakeholders.[79] The actual stakeholder of the Process Team is their organization that benefits from a successful business process.

The Process Team is to develop the model of the process as it is. But at the same time it should develop or work on their process itself (or on their working habits)[80]. For this to work out, a Process Team should include the mangers responsible for the work performance. The Process Team is also at least partly responsible for implementing a changed process and/or IT system into the organization.

Because these process participants, embedded in their working context, are no developers or team in the usual Scrum sense,[81] they will and can probably dedicate only part of their time for meetings, interviews, or discussions – and may not be too fond of those activities. Connected to this is the fact that in contrast to the Development Team, there is a big chance that this team might not be geographically located close to each other. Locations within the same office or building would already be a very good initial situation.

---

[78] This may not be unusual for agile development, but chances for that are higher since the source of requirements is the work of an inexperienced team.
[79] From a classical software development point of view
[80] Self improvement/*kaizen.*
[81] They are in most cases a new team with no experience.

Face to face meetings and even daily sessions and discussions would, of course be ideal, but are unrealistic given the two facts mentioned above. Daily Scrum meetings to synchronize work are most often out of question. IT-tools (most commonly E-mails, telephone-conferences, or more advanced techniques such as shared calendars, maybe incorporated into online/virtual project spaces and similar things) will be needed to organize this team, and schedule requests, or initiated face to face meetings.

As a side note: time requirements for the Process Team will differ vastly from the initial phase of a project, when there may be a great need for information input (probably a lot of work and meetings to elicit information), to later stages when it is more testing and verifying (maybe a little less work).

The Process Team might grow and change over time, because further participants or roles in the process are uncovered and knowledge or insight is needed from an according person. Initially during the starting phase of the project there even may be only one or two persons on the team, because at that point it could not be decided whom to involve.

Change does not automatically always mean expanding the team: for each role in a business process there usually is more than one real person that does the according tasks. So sometimes the organization of the customer needs to decide which people need to get involved. Because of that, part of the explorative idea is also to dynamically adjust the team members of this team.[82] Answering this question of team size and members should be a central part of initial planning sessions as well as planning for each iteration/Sprint.

What or how the team does this planning and other work will be coordinated by two further team members:

As any normal Scrum team the Process Team will work with a Product Owner, who will keep track of the tasks to do, and will have the final word on the project.

Finally the last member on this team will be the nCEA Master, who will serve as a guide for the Product Owner through the project and who will have a similar role as a Scrum Master does.

Normal Scrum teams are self-organized. One restriction for nCEA is that this will partly be limited for the Process Team, not because members are not allowed to coordinate issues on their own and schedule their own meetings, but because as an inexperienced team and with having other duties it is simply unrealistic to expect complete self organization and necessary discipline. Product Owner and nCEA Master will most often function as coordinators and initiation authority. However, they should not abuse that power and still try to give all participants good insight into the events that transpire (if they wish to).

The Product Owner:

The Product Owner mostly keeps his original role, but with slightly different toolset.

---

[82] It is also an option to form sub-teams for a sub-process that ‘report’ their work to the main team later on.

He is to decide on what needs to be done by setting priorities. The overall goal, of course, is to make sure the product – as described in the beginning of this section – is finished to a degree that it can actually improve performance of the organization. This includes the supporting IT system. Consequently he should also function as the Product Owner of the Development Team, because his knowledge and gained ability to understand the process is part of the *product* – and is meant exactly for that, the development of the IT system.

To achieve his goals, of course, he must be authorized to make decisions concerning the process and be allowed to direct resources. But to able to make educated decisions he should be aware of as many aspects of the process as possible. This requires him to be part of most interviews and discussions (and arguments) if possible. At least he should always be updated and kept informed on results and changes in the model, or possible problems or improvement ideas. His job, so to speak, is to learn the process and the model – meaning learning how to relate the model to the actual actions in real life, and how to translate that knowledge into changes and requirements.

This task can of course be split up between more than one person. There could also be supporting staff for the Product Owner. In cases of large, split-up processes a set-up with sub-teams and also sub-Product Owner could be possible. But as in Scrum, one single Product Owner should and will have the final word. To speak with the words of [ScSu10] *"The Product Owner is one person, not a committee. Committees may exist that advice or influence this person."* but nothing more.

The Product Owner should consider this position to be his main engagement during the project, and be dedicated to it, if possible, a hundred percent – unlike the rest of the Process Team who in most cases will have to do their regular work besides the project. Ideally though, the Product Owner should come from the same working environment and she or he should have work experience at least in parts of the process or processes..

Since the Product Owner should come from customer side it is unlikely that – at least at first – he is used to either the general ideas or the tools, and methods required to effectively run an nCEA project. To support him during his exploration there will be an nCEA Master.

nCEA Master - the Guide

The nCEA Master in the first place takes the role of the Scrum Master for the Process Team. In this position, he should be fairly experienced with the aspects of process modeling, analyzing, and improving, but also versed in e.g. people skills and human interaction, or the tools and ceremonies proposed for nCEA. His task is to guide the Process Team and Product Owner on the exploration of their own process, and show and explain to them what is possible or necessary. He is the expedition guide that may not be used to that particular jungle, but has experience with the tools and jungle exploring – to speak in the terms of the metaphor in section 1.5.

Scrum teams are usually well attuned to working with each other, to using tools, or to commencing and using 'Scrum ceremonies'. A Process Team in this function is usually a new creation. So the people in it may have been working with each other for a long time, but not in the sense of the working on their own processes. Because of that the nCEA Master will probably need to explain, teach or convince process participants of

the tasks at hand. He may need to introduce them to process centered thinking, and advise in matters of how to do them, or indicate steps that need to be taken.

The focus here lies on “advising” – the Product Owner has the final word.

The nCEA Master should also be the modeling expert of the team – drawing the map so to speak. For once he will (or should) probably be there most of the time when process information is exchanged or discussed. Secondly modeling needs to be done with some tool or notation, which the nCEA Master should know, but the other team members probably do not. Thirdly he needs to know the model by heart in order to e.g. explain, if questions arise, or indicate different process options to the team that might be better (if they themselves cannot perceive any). There is no one who knows a drawing or model better than the person who drew or created it.

Of course, the role of the nCEA Master could also be split up – for example into a separate modeler, and a kind of communications officer that is responsible for passing and structuring information between Process Team members, and who manages communication and knowledge transfers. But this would come with drawbacks: work done separately needs to be coordinated separately. Consequently, the likeliness of confusion grows due to the increased possibility of contradicting directions from different nCEA Masters. And of course more people do cost more, and are not really necessary here if the nCEA Master knows his tools – or at least how to get them set up from the Development Team.

Additionally the nCEA Master is supposed to fulfill another function:

To bring in and coordinate external experts with domain knowledge on process improvement (muri, mura, muda reduction) if a possible application of such experts is has been identified. To be able to do that the nCEA Master should at least be familiar with the concepts and know where they could possibly be applied. The nCEA master will thus be the stable external face to the process natives, in contrast to maybe more frequently changing ‘experts’ he is part of the team and needs to work for them.

Interaction between Teams

The general team set up should look something like this:

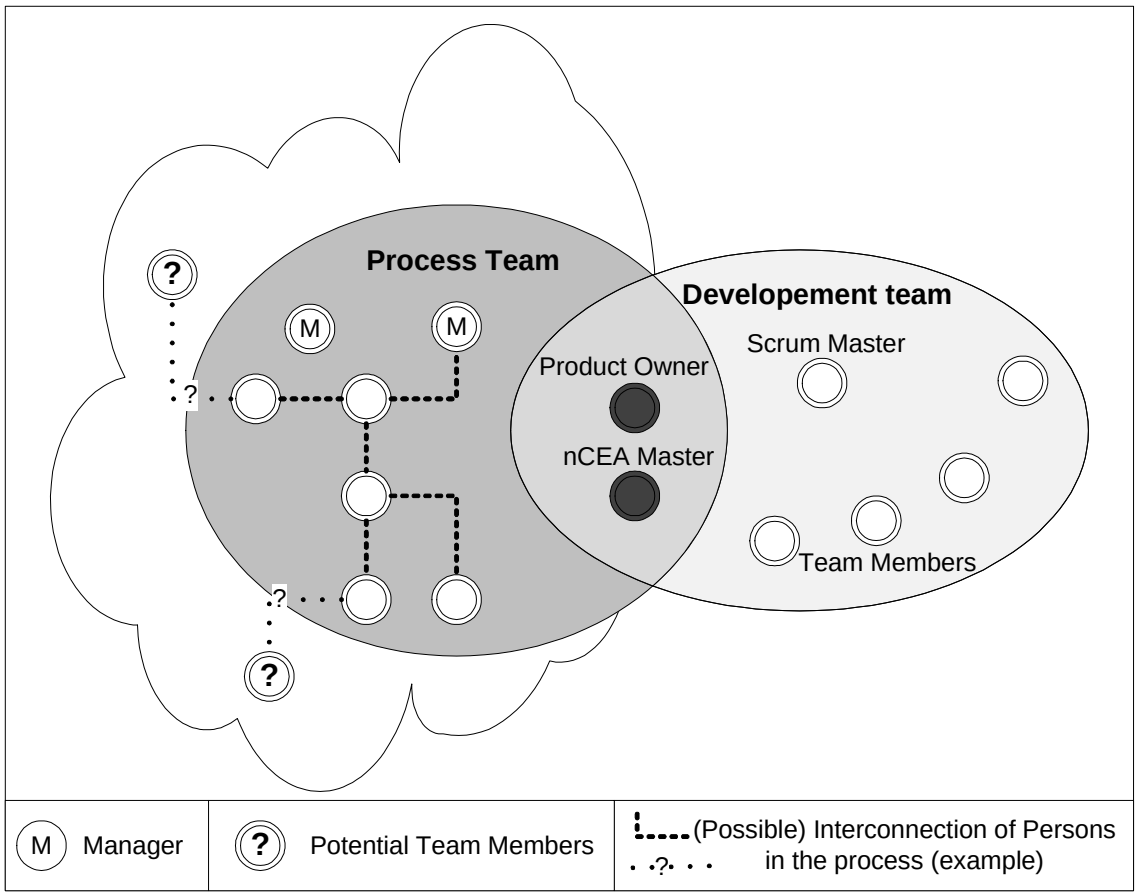


Figure 24: Schematic of nCEA team set up

As has been said in the description of the Process Team, those people will also be involved in their own day to day work, distributed geographically, and cannot be on standby for questions by the Development Team. The Product Owner and the nCEA Master, as members of both teams are meant to coordinate and transfer information between the two groups, with the model as their main tool.

Exploration/gaining understanding of the process, and implementation of the IT system need to be done simultaneously if a certain speed and actuality of development is supposed to be reached, and in order to have at least somewhat up-to-date digital supported processes and models. This set up of the project provides the framework to make this possible.

This description also contains the implication that technically, such a project can never be really finished, since processes constantly change and thus IT-systems or at least the models underlying them need to be updated constantly. A preset project frame usually gives a deadline for such projects. Within the given time, the overall goal is to simply get the best possible solution.

Nevertheless, it is possible to continue this cycle if the project is extended or if it is taken from project status to be part of a business process within the organization itself: a continuous improvement process, with set intervals, but an open end.

Sprints for Process Teams in nCEA

A regular timeframe or organization in time boxes is *"the backbone of any agile project approach"* [Oest06]. For Scrum, these are the Sprints and bring order in the otherwise slightly chaotic work style. Especially for the more loosely structured Process Team the Sprints – with their repeating ceremonies – will be the anchor for the people to group around.

Cycle time for Scrum Sprints can vary between one to four weeks. What can be defined as a good cycle time is vastly dependent on the actual organization and experience of the Product Owner and nCEA Master. For new Scrum teams usually a Sprint length of two weeks is advised as neither too short nor too long – this also applies here.

A Sprint starts with a Sprint planning, where the team decides on what needs to be done during the Sprint. For an nCEA Process Team this means first planning "when" (to meet); basically, team members need to commit to dates to do some things together.

The tasks that need to be scheduled come from the Product Backlog and, based on the decisions solely by the Product Owner, are set as to-dos for the current Sprint.

During the Sprint the product will be created. In nCEA, as has been mentioned before, 'product' means the symbiotic combination and use of model, IT-system, and context. The IT-system itself will be built by the Development Team in the usual Scrum manner, in a normal, incremental way. The model, the process (and its evolution), and the requirements on the other hand will grow/change or be refined by discussing them, again and again, but always with (hopefully) new contextual information or new versions of the IT-System in testing.[83] Progress can be measured in growing and changing model (or model version), the Change Log, and the Product Backlog of Development Team, that should fill over time.

As is usual, at the end of each Sprint there will be two meetings, that should have a fixed length and that all Process Team members should take part in.

Goal of the Sprint Review is it to inform person on what has been done or discovered during the duration of the Sprint, what problems or possible improvements have been found. Change Log (If used) as well as Product Backlog should be collaboratively discusses and updated during this meeting.

Finally, there can be the Sprint Retrospective where not the product or process is discussed, but the project, its organization, and where the proceedings of the Sprint are evaluated. This includes team formalities. E.g., if people consolidated during the last sprint or who have otherwise gotten temporarily involved should be formally introduced into the Process Team.

As a post scriptum there can also be Sprint Review Reprise. There are two teams with their own Sprint Reviews, and Retrospective meetings. But at least two members (nCEA Master and Product Owner) are the same on both teams, so the

[83] There is no need for discussing things if nothing – no new information, no new persons, or no sudden contradictions or problems - has changed since the last Sprint. That would be futile and fruitless.

aforementioned meetings simply cannot happen at the same time[84]. They should be close to each other in time though, as Sprint cycle time of both teams should be synchronous. It is not unlikely, that during the Sprint Review of the second team, information that could influence the work or plans of the other team[85] turns up. To discuss those, the team that had its Sprint Review before the other can hold a Sprint Review Reprise meeting, much shorter than a full Sprint Review, in order to get updates or discuss the new matters, or update the logs.

Afterwards the cycle starts again with Sprint planning for the next Sprint.

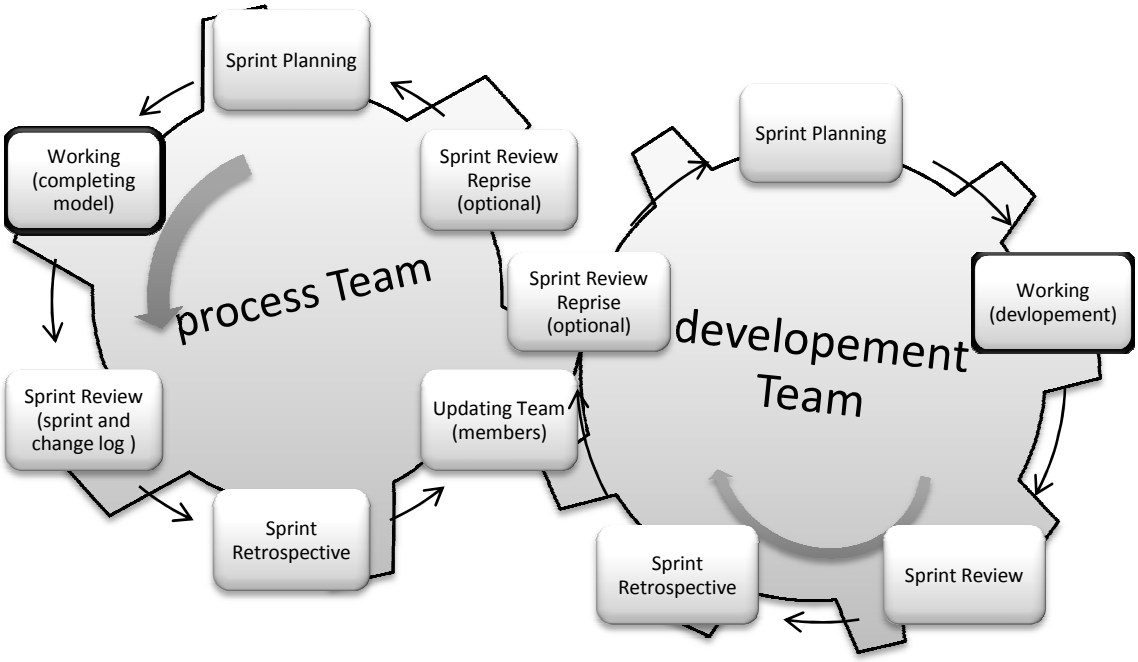


Figure 25: Interaction of Sprint cycles for Process Team (left) and Development Teas (right)

Sprint Review, Retrospective and Sprint planning meeting should be done separately – maybe a day apart, if possible, in order to give people some time to think about what was done and what can be done. If the usual length of such meetings applies, having all three meetings for both teams each on the same day is impossible anyway.

Product & Sprint Backlogs

In Scrum "*the Product Backlog is a prioritized list of everything that might be needed in the product*" [ScSu10] .

In the beginning there is simply one item on the Product Backlog that will stay there the whole time: the improved process. The sub-items to that point are by default: creation of an adequate model of the process (with a purpose), the eliciting of requirements for IT-support-tools and their validation, and changes to the process structure itself. These

[84] Or at least that should not be done. Having substitutes for these two roles is a possibility but requires greater effort in coordination and preparation and won't have the same quality.
[85] Ideally important information was incorporated before by Product Owner or nCEA Master who should constantly be up-to-date.

sub-items will be present in every sprint. They will be further broken down into more concrete, executable units and other items beyond can be placed as required.

During the sprint planning meeting the items on the product backlog are to be translated into work tasks for the Sprint backlog.

But what will be done at all? This may sound silly, but it will be the interviews, meetings, and discussions during which information about the process can be exchanged, be elicited, and be captured in the model. Other examples may be the setting of dates of when certain changes are implemented or evaluations. They all will function as the representation tokens for knowledge, information, clarifications, model verification, or feedback on the IT-system, which are the actual "*product"* or are part of an adequate product. In the end it is only this human to human interaction that is actually the core part of any agile development activity following [RoSh10].

Deriving items for the Product Backlog of the Development Team should work as follows: while discovering their own process the process participants will come up with requirements for the process (or at least their own work tasks). These may translate into ideas or changes to the process, but in case of wishes for the IT-system[86], are meant for the Development Team and need to be on their Product Backlog. It is up to the Product Owner and the nCEA Master[87] to transfer these items. Before transfer, though, the root cause for request should be clear in order to avoid bug-fixing problems or errors that are actually rooted much deeper at the process level.[88]

The opposite approach, of course, is also true, and even standard, as anything the Development Team produces will find its way back into the logs of the Process Team. New variations or feature need to be evaluated and feedback given (for which according meetings or discussions need to be scheduled).

The following gives examples of what kind of items can be on the Backlog of the Process Team. These items are in no particular order. It can also be an example for 'meta'-items, while other may actually be sub-items or be similar to each other:

- The model
- Meetings of (at least part of) the Process Team
- Sessions with optimization experts for the process
- System evaluation (system creation is on the backlog of Development Team)
- Process and requirements elicitation
- Interviews (single persons)
- Discussions/group interviews
- (partly) Model presentation and verification session.
- Clarification sessions
- System or feature feedbacks

---

[86] E.g. automatic retrieval of information from databases like address books, or automated processing steps

[87] Because these two are most often the only cross-team-members.

[88] In one of the interviews done for this thesis an interview partner told such a story, where participant wanted a function for the workflow tool they were using to support the process, in order improve their overview at a certain point. Later and more by chance it turned out that the actual cause for that which lay within the design of the process model and not in the tool itself. Modifying the model was a quicker and better way to fix the problem. → "five times ask why"?

- New system functionality presentation (training + feedback)
- Problem analyzing.
- Trials of new ideas of process handling – there is a time frame and duration, and measures to evaluate the idea
- Evaluation session of previously made process changes.

Change Log:

The change log is a subsection of the Product Backlog of the Process Team. The ideas is simply to keep a list of previously made changes to the process (be it in the process itself, the IT-system, or a mix of that). These changes are on tryout for certain duration of maybe a few sprints. At the end of this duration their effectiveness needs to be evaluated. If this evaluation should turn out unfavorable some action even should be taken back.

The idea for the *Change Log* derives from the considerations in the first section of the thesis, namely that it is a common mistake in handling complex systems not to go back and review and evaluate the consequences a change or action has brought. Not every change may lead to a desired improvement.

So for every change agreed on in the process (or a collection of smaller changes) that at least has a medium impact potential,[89] there should be an item (e.g. a characteristics note-card) that stores information on the change:

Briefly note down:

> Who are the affected subjects/persons?
> What was the change?
> Who was involved with introducing the change?
> What were the reasons for the change or idea?
> What where possible (estimated) consequences?
> Ideas and strategies how to measure consequences or effects of the change:
> Note down a trial period:

During one of the future Sprints an according change evaluation meeting among the people involved should take place.

The Change Log is mainly meant as a tool for the Process Team to keep an overview and also to force them to review their actions taken.

The Daily Scrum for the Process Team

Since daily main work for the Process Team members is probably different from project, nor are they probably located close together, there will be no general Daily Scrum. Now the main purpose of the Daily Scrum meetings is for the team to get an intuitive and continuous feeling about what the other persons in a Scrum team do (in order to inspire their own work or derive ideas or synergy effects) – which will be missed. Getting that daily information about the project and its progress is vital for team building effort. And meeting every two weeks with the whole team – even if only virtually – and otherwise

[89] Minor or cosmetic changes without maybe omitted, though it sometimes

just wait for some people to come by will not really give participants the feeling that they are involved.

In order to at least emulate the purpose of Daily Scrums the idea is for every action or item on the backlog done, one of the persons involved (not nCEA Master or Product Owner![90]) could write a short summary or protocol roughly maybe the length of a "Twitter feed" or SMS – very short[91]. The idea is after each such meeting or other 'item' done to answer the questions that normally would be answered during Daily Scrum:

- What was accomplished in the meeting (original: since the last Daily Scrum)?
- What will need to be done before the next meeting, or which tasks are next?
- What obstacles have been found or need to be cleared up?

These short protocols then should be distributed to all Process Team members via a channel of choice (e.g. E-Mail or bulletin board in a virtual process portal). Skipping reading over these short protocols should be allowed, but should be done actively[92].

Sprint review Reprise vs. Scrum of Scrums

On the first glance the concepts of nCEA Master and Product Owner being part of two teams and the Sprint Review Reprise is similar to the concept of "Scrum of Scrums"[93] that is meant to coordinate two or more Scrum teams that work parallel to each other,. while at a Scrum-of-Scrums members of each team come together to report, plan and coordinate their actions.

The difference in nCEA is, however, that Process Team and Development Team are not really working in parallel, on separate parts of the same product, but entwined. The 'product' of one team potentially affects the work of the other team much more profound and directly than with parallel normal teams.

Together, Process and Development Team form a unit that could be coordinated with similar units[94] to manage larger projects.

Burn-down-Chart

The main goal of burn down charts is to get a visual aid or representation of work progress achieved so far. It also is supposed to be a motivator as it should show how much progress has been made by the team.

In reality it is often placed on office walls or other places were the whole team can see them.

In the cases of nCEA, the usage of burn down charts may not be too practical. For once, there is probably no single location for the Process Team to hang the chart on a

---

[90] If it is an task that involves not only the two of them. Those should be also involved
[91] The idea for information given in Daily Scrums is to stay brief and do it quickly, as such it should be kept with the short protocols in order to not overburden team members with unnecessary information
[92] The main idea with that is, to force people at least to acknowledge that something was done instead of simply not knowing anything at all.
[93] (see [Larm09] for the concept)
[94] Not considering special cases were maybe one development team serves with more than one (sub-) process team.

wall where all team members have access to it. A digital alternative virtual/online project space representation may be required if a chart is wished for.

More importantly, testimony of experts states that the effort needed to in such projects is often vastly underestimated in the beginning. Also many actual tasks can and requirements will be discovered during the process. Without at least roughly adequate estimations for remaining effort, the 'hill' on a burn down chart is very likely to grow or stagnate for a long period of time, instead of actually getting smaller. Thus the positive effect of a 'work done' visual may be negated or even inverted.

This would hold especially true in cases were a project would be converted to a continuous and lasting nCEA process.

Rarely to be alone

A project as described here of course is an ideal only. It is supposed to include all steps from process documentation, over improvement and optimization, to IT-system integration. It is also assumed here that there is only one relevant process[95] to be worked on by the teams.

In reality though, this will rarely be the case – except, perhaps, in prototype or test projects. There are usually many processes with only partly overlapping circles of participants. Also many involved people in general will make the situation more complicated, but at the same time possibly allowing for synergy effects in time or process improvements.

In those cases tools to scale up Scrum or agile methods and handle the according problems that come with parallel or large scale developments should be applied. For further reading into the matter the reader can refer to [Larm09] and [Larm10] as a good reference on the topic if the need should arise.

Future Prospects

nCEA is intended to be compatible with the average working environment in businesses. Thus the nCEA Master functions as modeler, and expert in conducting the project – which, strictly speaking, he is not supposed to be, in the same way that a Scrum Master is not supposed to work in the manner of classical project manager with a lot of directing power.

A future prospect is that nCEA teams do the modeling on their own. The idea is that the more experienced a team gets, the more the nCEA master will turn into a mere coordinator and collector of information rather than a guide.

[95] Or at least a single process context, where the participants of several process are identical.

4. Testing nCEA– an Adoption of KreditSim

KreditSim

KreditSim is a role-playing game that simulates the processing of credit or loan application within a financial institution. It was designed originally by the Process Lab Research Center of the Frankfurt School of Finance & Management[96] for participants to experience and practice the application Six-Sigma- Methodologies on fictitious business process out of the financial service sector (instead of the manufacturing sector where Six Sigma originates).

During the game the participants take different roles or stages mainly in the back office of a virtual local financial institution specialized in financing construction projects. Jointly execute a business process with the goal of fast and correctly process loan application in order to satisfy customers as well as the banks own financial expectations.

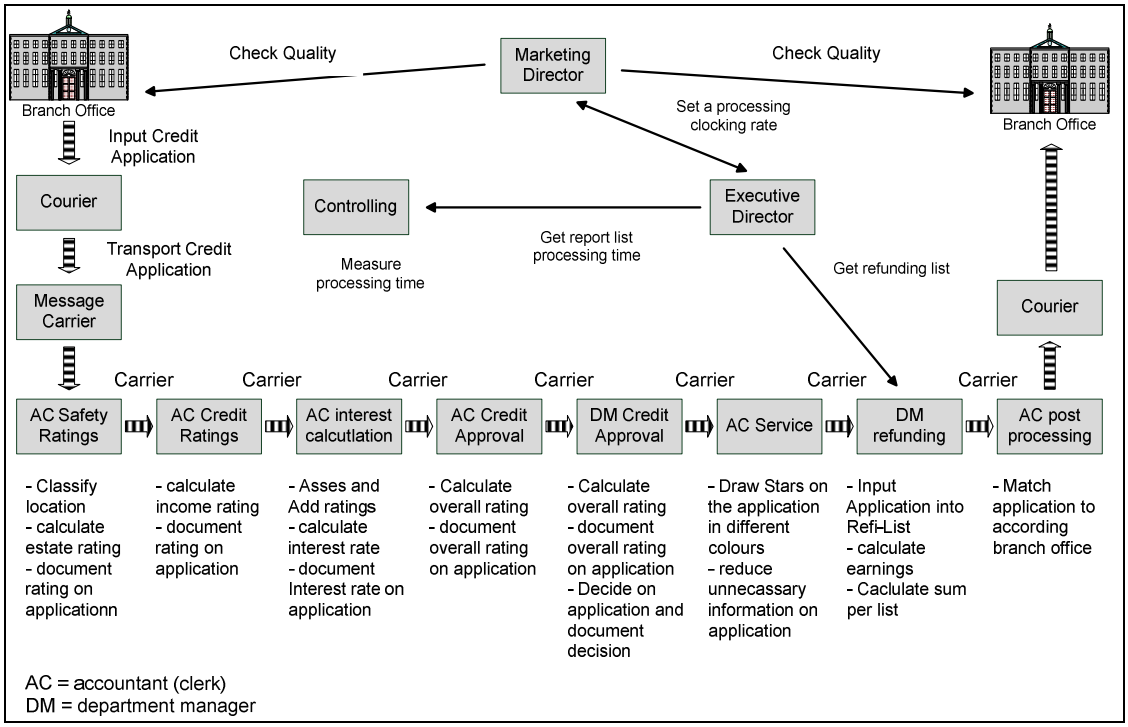


Figure 26: Overview of the KreditSim game. (C) Research Lab

The game is supposed to be played by a group of up to eighteen people in one room during a seminar. During the course of the game, participants will slowly realize that the original process has quite a few flaws and are supposed to improve the process by applying certain procedures of analyzing the business process, measuring its performance, improving it, and checking the results.[97]

At the University of Karlsruhe, this game was already experimentally used in connection to S-BPM methodology in form of the Metasonic Business Suit during the spring term of 2010. Instead of a pure pen-and-paper approach, the experiment

[96] Official web page [Proc10]
[97] Using the concept of the DMAIC cycle which is akin to the Deming Cycle referred to on page 18 of the thesis.

participants used an IT system running on the “original” process/model. Afterwards, the tasks were analyzed as usual and an improved process model was generated, introduced and via game simulation evaluated. All that was done within one day.

In that case, the seminar team responsible for modeling and conducting the experiment had prior knowledge of all the game data and process description of the game for every single station of the game.

Result: it was possible to quickly and efficiently play through the game, evaluate it in quasi real time (because all necessary data was already measured by the Metasonic engine). The game worked even with significantly less supporting staff (like message transporter, or performance measurers). As intended by the experiment, it was possible to significantly reduce errors and speed up the processing.

Adopting KreditSim for nCEA Testing

To test nCEA in such laboratory setting, certain deviations from the game are necessary.

The duration of the experiment will last longer than one or two days, possibly two to four months.

A seminar of people playing the roles of the bank employees and a modeling team (2-4 people) that will do the actual work will be required. They will function as Development Team, Product Owner, and nCEA Master[98] in the nCEA framework.

At the beginning there will be an initial playing session with the original process in order to train the experiment participants for the roles they are playing. It does not matter whether it is the original pen-and-paper approach or an IT supported game.[99] Since there will be not too many of these sessions and the participants are in contrast to real bank employees not accustomed to their job, at the end of the first session they should be asked to take notes on their personal opinion or ideas for the process to keep for later meetings and discussions.

Ideally the first playing session is not conducted in a single room in order to prevent the role-players from getting too good an intuitive and spatial ‘gut-feeling’ of how the process works, or gain too much opportunity to review the process among themselves, in order to simulate the lack of overview in the real world. For the same reason, they should not see a depiction of the overview chart. Equally, all other information about the whole process is kept limited and to a minimum on a must-know basis. Everything else needs to be discovered. The goal here is to artificially split the process knowledge up among many persons.

Especially the modeling team must not gain any information prior to exploration about any parts of the process and are best left out of the initial playing session.

---

[98] Since in a university setting it is unlikely the participants in the modeling team will have prior knowledge of the S-BPM methodology or others, a trained helper from experiment-staff may need to support them, especially on the role as nCEA Master.

[99] If an IT-supported version is chosen (especially if it is also an S-BPM version), process participants should be prevented from accessing the visual depiction of the underlying model.

In the function of the Product Owner they should be provided at most with the overview or general idea of the process (e.g. with the rough overview chart depicted in Figure 26). They also may be informed about performance of the initial game as the fictional motivator for an investigation and improvement project.

Their overall task will be to build a model, implement it in an IT-System (in this case the Metasonic Business Suite) and improve the process together with the participants. So in a final session the game can be repeated with the experiment participants to, hopefully, run more smoothly.

During the course of the experiment, in order to explore the process and get the information that is split up among the other participants of the experiment, the team is free to try to coordinate meetings with the others over whatever medium they prefer. Because the other participants will have other duties in real life, this should (ideally) not be easy and overall sessions with all participants may not be possible due to conflicting schedules and lack of meeting spaces – which should simulate the problems of real world situations.

As nCEA is time-boxed, an nCEA sprint meeting should be held every two weeks in order to inform the supervising experiment staff of progress, discuss problems, and plan future steps.

It is open to the creativity of the participants which tools are to be used to keep track of the sprint and Product Backlogs, other process participants (or similar information that might be useful for all participants) , though tips and ideas can be given.

Evaluating the Experiment

The experiment can be evaluated in several ways.

Since in the previous experimental settings, S-BPM based models of the process were produced, too, it is possible to compare the explored model with a model that was created under complete knowledge in regards to completeness, adequacy, or style. Also comparable is the process performance that was achieved and measured in the previous experiment, before and after process improvements. Other points of analysis will be whether all information was discovered, why something may have been overlooked, and what maybe was done better.

Furthermore, the development process should be documented by the process participants and reviewed at every Sprint meeting, with the analyzing focus on what went well, what went badly, and what potential reasons for such observations were. The protocols than can be analyzed to find possible problems or misunderstandings of nCEA during the course of the experiment.

Finally feedback in form of expert interviews with all people involved should be gathered in order to get opinions on the group work, coordination, and to get additional ideas.

5. Summary & Outlook

.

Following [Müll10], every study and every final thesis written in university should be concerned with one of 'the big questions' in life. The list of these pillars of science or rather topics for the future reads as following: *energy, life, health, information, social systems, and cultures*

If one was to place this work in that system, it would be akin to a bridge spanning between 'information' and 'social systems' columns.[100,101] It is a cross-over work, located close to the pinnacles were those pillars touch 'the sky of human every-day life'. It brings together and binds ideas discovered further down the pillars where research is conducted, drilling down "within" each column into the never ending depths of the fields of science – basic research and foundation works.

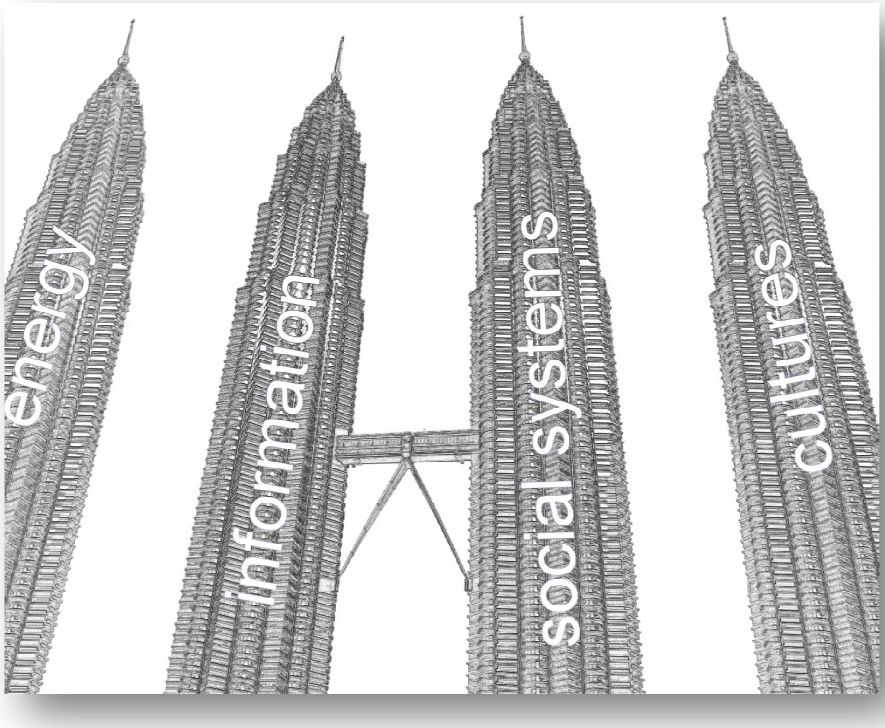


Figure 27: A bridge between the 'big question' pillars

The basic problem to be challenged was the question how As-Is-business processes can be mapped, modeled, measured, and via those actions conveyed into a continuous improvement process.

[100] In the system of the classical disciplines of science (represented by e.g. the department structure of universities) the thesis would cross in between IT, psychology, organization or management science and partly sociology.
[101] Of course this work is not the only, nor is it the biggest bridge as there are numerous tries and attempts (born from necessity) that span the gap.

The possible solution given in this thesis is the explorative application of Subject oriented BPM approach, which is named nCEA – the natural Context Exploration Approach.

The concept conveys ideas from several directions. The core is the original process approach from the S-BPM pioneering company Metasonic with own their prototypic process approach and the agile development framework SCRUM, that are combined. Additionally research results or ideas from several different fields have been incorporated. Namely from psychology, system theory, or Japanese manufacturing philosophies in the form of the Toyota production system with their *kaizen* pull-principle

Beside the general concept to combine the ideas that were mentioned, the key concept derived in the thesis is that of the 'personal process space' as the most pragmatic[102] unit or module a business process is actually made up. These are the units of that are to be explored. Closely related to S-BPM's 'subject' concept, at the center of each personal process space stands a human being as one processor in a process.

Practical experiments with S-BPM conducted at the KIT during the summer term of 2010 have shown that this kind of setup or mind frame is indeed very useful for modeling. It allows for the complete modeling of a business process in all its variants in a very easy yet fast manner. So easy in fact that even end-users may quickly get to a point where they might be able to modify their personal process spaces in collaboration with their neighbors.

nCEA was derived to give exactly that idea a constituted form, that allows to anchor such continuously improving methods within the organization that runs the processes.

Further Research

To return to the bridge metaphor for the thesis: Of course this bridge can and should be improved by testing its stability and thus it's usability in the 'storm' of reality (e.g. via an evaluation as proposed in Section 4, or via challenging experts on their opinion). Furthermore, in the fields of either pillar (or beyond that) more 'anchor points' for this bridge can be found to stabilize or even modify the shape of the bridge. One such example would be differentiation into the finer details of group interviews vs. one-on-one sessions, or considerations into what interview style is how practical in context of different cultures of the participants.

If groundwork research is described as drilling down into the bedrock of the pillars, then a bridge spanning a gap is not a very good starting point for further drilling – as a starting point of research itself so to speak. But at each side of the bridge inside the columns, a multitude of ideas still need to be evaluated and questions need to be answered, especially concerning the topic of S-BPM, the origin and starting impulse for this thesis.

On the (in-)side of social system (or psychology), the open core question is whether the hypotheses in the 'Advantages of S-BPM Models'-section are true, especially the stated advantage of understandability. That hypothesis is based upon older findings

[102] Most pragmatic in the sense that there are many possible ways to split up, or separate a business process for many different reasons, but the PPS with human beings as the core element is the most useful in terms to handling, understanding, or exploring.

and personal experience, but research still can and needs to be done to verify those findings, and to determine under which preconditions, or up to what dimensions of systems they hold true – or how they need to be adapted following the abduction logic. This can be done in extensive psychological experiments, with set-ups involving several groups of people (with e.g. different background, pre-knowledge, experience), comparison groups, standardized process descriptions/models in several modeling notations, and standardized set-ups or situations where those process models need to be evaluated, communicated, or created by participants.

In the field of organizational or management science, an open topic could be how nCEA can be applied or works within existing ITSM[103] or IT-governance frameworks that go beyond governing the simple project progress - such as e.g. ITIL[104], MOF[105], or Cobit[106]. These management frameworks are meant for larger organizations and are not only concerned with the creation, but also with the running and maintenance of IT systems and complete IT infrastructures, covering their whole lifecycle. They usually are comprised of several standards and best-practices that nCEA (and other agile approaches) may be contradicting or clashing with (or not). Further thoughts and evaluation with experts on that topic are needed to discuss possibilities, options or restrictions such integration of nCEA would face.

In the same direction lies the issue of scaling up S-BPM models (and indirectly the parallelization and scaling of nCEA projects) that might be worth further investigation. nCEA clearly is a bottom-up approach, and focuses on the natural unit of business processes, the human being. On the other, hand the overall processes of large businesses may need[107] to be described on a very abstract and high level. The time it takes until a model can be aggregated to such high level may be too long. As was said in the 'Top-Down vs. Bottom-Up' section, no pure approach can and should be the solution, as either approach will consume much time and not lead to a favorable result on at least one end. A possible solution to that may be two different modeling systems that are only theoretically connected or loosely coupled.[108] One at the 'bottom' for the process modeling as described within this thesis, that will produce usable and practical IT system to support processes, and a second one for modeling the business (organization),[109] as an information tool for upper management.

[103] ITSM = Information Technology Service Management
[104] ITIL = Information Technology Infrastructure Library [ITIL10]
[105] MOF = Microsoft Operations Framework
[106] COBIT = Control Objectives for Information and Related Technology [ISAC10]
[107] 'Need' in the sense that upper management is usually in need of a mind-model of their company.
[108] Loosely coupled in the sense that information may still be passed 'up', but especially controls or directions are processed downwards only in the flesh and not in the system. A change in the 'upper' business model does not lead to direct change or disabling of functionality in the 'lower' system, while a change there may alter some indicators, but does not crash the whole system.
[109] 'process modeling' vs. 'business or organizational modeling' so to speak. Alternatively it could be said 'together with' while being aware of the slight differences between both concepts.

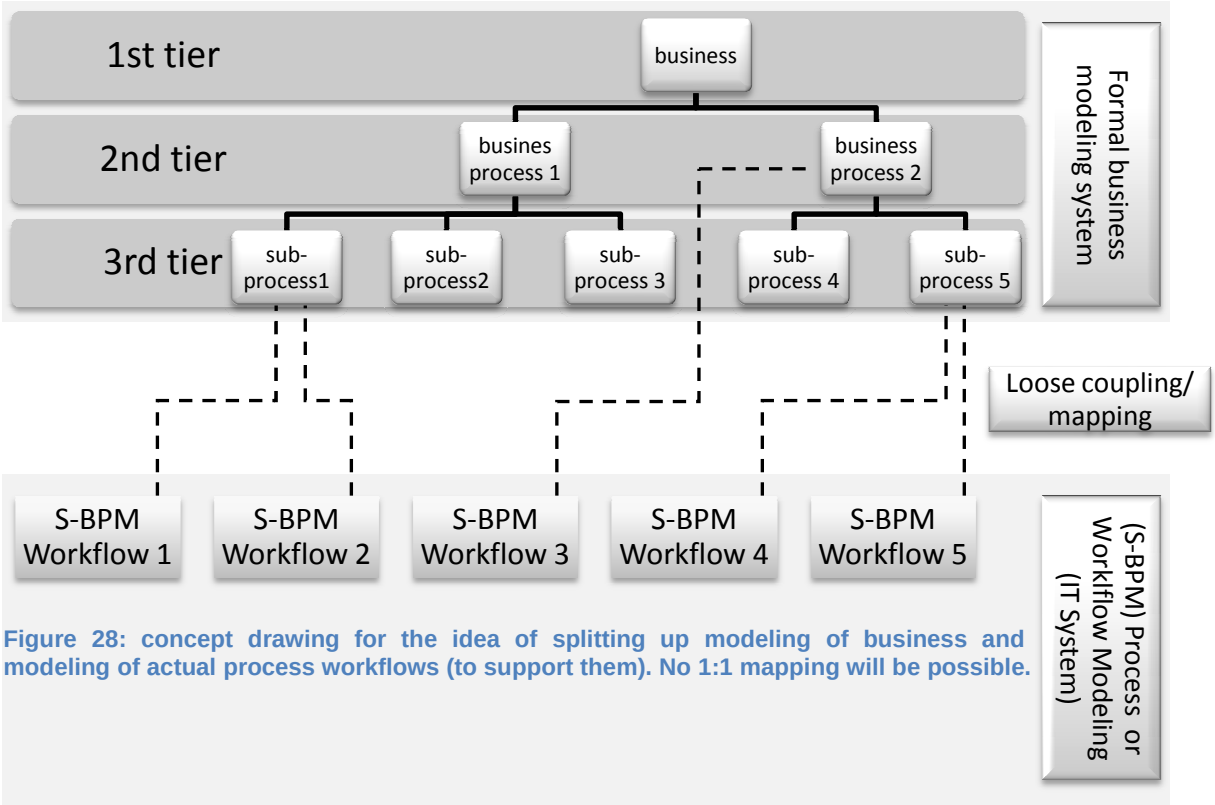


Figure 28: concept drawing for the idea of splitting up modeling of business and modeling of actual process workflows (to support them). No 1:1 mapping will be possible.

In that way, the mistake of trying to construct one big rigid system model – that cannot be used until it is complete – can be avoided. And continued improvement at the bottom in the kaizen spirit is not subdued in a rigid and dogmatic environment, while at the same time it does not deprive upper management of their requested company overviews and information.

Interesting questions here would be, where such a 'break' is best facilitated, how exactly information will or can be passed across such an artificial 'gap', or what consequences or acceptance such a division in a BPM tool might have. Or possibly whether such an artificial binary split is actually necessary at all if a revolutionary design or modeling technique can be devised that maybe splits up the whole system into autarkic operational sub-units.

Finally, further research questions await the topic of S-BPM deep inside the information column:

One question here is the formal definition of a 'subject' of S-BPM notation and how it can be represented. One such interpretation of a subject may simply be as an interpretation as sub-partition of graph that is defined only by a semantical context. Other possibilities or directions may be related to interpretation as units in dynamic social networks, or on the more technical side, the interpretation as communicating abstract state machines in combination with the 'explicit' representation of different 'processors' in the processes.

Another topic of particular interest in the IT Field is the simulation of business processes[110]. Research such as [Puhl06] has shown that the Pi-calculus of Robin Milner [Mil04] , a calculus devised to formally model and compute mobile processes, can function as the formal basis of BPM. Yet the Pi-Calculus is not suitable for all simulation needs as there are several extensions to it in existence. One such extension is the so called 'SpacePi', 'a spatial extension to the Pi-Calculus' [JoEwUh08] , which adds time and space coordinates to the calculus, and is used to describe and simulate process in micro bacteria cultures and colonies. A superficial study into that field has inspired ideas to apply such a concept for business processes and to maybe even represent the idea of personal process space and human behavior within that formalism.

Finally, continuing the previous thought, it is interesting to investigate in how far the concept of 'personal process space' can be formally described and to what extent such a definition can be used in order to further help resolv or at least bring structure and order to the problematic and complex field of human and IT domain interaction, to the interface, where technology meets human live. How these two should meet and to what extent is, after all, the ultimate question and source of problems. This thesis has presented a solution to an aspect of that field of problems, but as the last section has shown, as always, there is much left to be done in the future.

[110] Because simulation can be the key to further improve business processes, by allowing testing and measuring of (collaborative derived) ideas to improve processes, which in turn can save cost, time and prevent frustration with process participants of possible problems can be detected much earlier that before.

Appendix A – Expert Interview Summaries

As research for this thesis several expert interviews were conducted. With person involved in business process modeling and the corresponding project management.

Interview with Dr. Albert Fleischmann:

Function: Chairman of the supervisory board of Metasonic AG, originator and initiator of S-BPM Methodology and tools, long time process expert and consultant.

General topic: general task and thoughts on process modeling and approaches (with focus on Metasonic S-BPM suite)

Interview language: German

Time and place: 30th April 2010 – Metasonic AG Rohrbach, Germany

Duration: 1:14 minutes

English Summary /interpretation of the interview:

- There are two approaches for modeling in general: top-down and bottom – up.
- Top-down: start with process architecture and refine the model down. Problem there: you build a ideal-type model for an ideal company (if such think is possible), but you can miss reality a long shot. It is deriving work directions from some far away general business directions. Connected to that: Hoshin-kanrin (aka policy deployment): methodology to break down business targets from top-to bottom. – this can go badly wrong
- Bottom-up: get persons from any deployment and pretty much listen to what they have to say. Write down for everyone what you do, and with whom you communicate. Put all the things together and consolidate problems. And together you can derive an idea what do we want to achieve together.
- Both approaches are possible. In reality a dual approach incorporating both is probably needed (“Jojo-like” approach)
- There is probably no proof what approach is necessarily better: Danger with bottom up is to get stuck in the given/current structures, because no one wants to change or give up old habits. Danger with top-down: you get something completely else from what was originally intended and end up with a giant change-management-complex (what tends to chaotic).
- Other description: business-process-reengineering vs. kaizen. It is a matter of personal taste.
- The thought pattern of giving directions from top to bottom may be a relict of Taylors organization theory. The division between organization and actual work. Though e.g. in manufacturing, where Taylorism originates, it is already seen that that is not perfect, and with business or administrative processes you need something else. Other methods
- The idea that the working processes can be self organized might be strange, but is comparable to a story: the chief editor for Brockhaus (German encyclopedia) could not imagine that via the Wikipedia a qualitative encyclopedia could be

compiled. So why not try something like: wiki-process-management – as a wild idea. But it is not really possible to just tell people to organize themselves

- A synthesis of both approached might be the best: given a rough and flexible process structure from top to bottom.
- A workflow engine (and consequently the model used to run it) is needed only for the bottom, for people doing something (the actual work). Anything above the execution level is not really necessary.
- The whole process of finding and setting goals or defining KPIs etc. serves in the end one purpose: so people can or will do something. If not one can scrap anything before that.
- Without knowing what one is doing (in business) one cannot really manage. Otherwise one will get swindled and lied to from the people on the bottom. And relatively easy: not even intentionally, if one simply cannot understand subordinates.
- On Organization and understanding organization: you cannot control everything – former communist block is proof to that, where everything was managed and controlled centrally – and failed spectacular. The big experiment was cancel after 70 years. One simply cannot control everything.
- There are management approaches that do not consider organization to be rigid: cybernetic or system thinking approaches where everything in an is described as feedback loops and the like within organization.
- Derived from such considerations Fleischmann is fascinated by the works of Luhman (German Sociologist) who describes “communication” as the smallest unit of organization – which is akin to Fleischmanns ideas.
- On Models: Model of course can never be perfect. It means leaving out details, and when doing so you automatically make mistakes.
- General Problem of BPM: what detail level or granularity do you use for a model. If one takes communication as the smallest unit you automatically get to the S-BPM methodology. The granularity of the process is determined by the involved subjects – the information processing processors (according to Luhmann) And in the end the smallest unit is the human in the context.
- This consideration can give a very good idea about where to stop (modeling details of a process). With emphasis on the “can” – not must.
- For every subject (in hindsight to Metasonic jPass) can the granularity level can be determined individually.
- Because breakdown of task among subjects is rather hard, modeling should start with thoughts on what subjects are involved.
- There is no algorithm or automatism, that finds “the best” solution to split up task (among)
- In the end every process can be considered a massive network of subjects. Everything else (other considerations) are just tries to manage the masses.
- One cannot learn complex circumstances in a linear way. Humans can only think, and communicate, linear or in sequences – at least consciously (base on the book “*On intelligence*”).
- The world of course “happens” or is ‘parallel’, humans though can only think ‘sequential’. So because “we cannot fly” but “we need to fly”, humans build tools to support them. In process case we need to structure and bring order to the parallelism to process it sequentially.
- S-BPM modeling allows to start sketching, a process while talking to a single individual, without the need to consider the whole process at the time of learning.

- On process modeling: two other approaches: modeling by constriction or “design by restriction”
- Modeling by construction is simple: only what is given is allowed an can be done
- Modeling by restriction: everything is allowed and possible from the beginning (everyone can communicate with everyone else), everything is generic. And during modeling the generic allowed actions are limited further and further.
- It is kind of a process description where one starts from one universal process and restrict it more and more
- Question can one and if yes how can one combine top-down or bottom-up approaches with modeling or design by restriction.
- Basic idea of modeling processes: a special template to help and speed up future processes of the same type.
- The origin idea of the “universal process” is what is happening anyway: communication via E-mail and telephone, possibly between everyone. People simply do not thing about it.

A larger part of the rest of the interview ended in a lively discussion about modeling-by-restriction.

Interview with Christoph Schulz

Name of Interview partner: Christoph Schulz

Function: Customer Consultant at Metasonic – company

General topic: modeling experience in projects (wit S-BPM)

Interview language: German

Time and place: 10th June 2010 – Metasonic Company facilities.

Duration: 40 minutes

English Summary /interpretation of the interview:

Christoph Schulz has extensive work experience in the field of S-BPM methodology, with modeling business process in projects with customer interaction. All his statements were in reference to projects with Metasonic S-BPM Suite and methodology.

- (With at least one customer) initial process documentation was on power point slides, separated into several sub processes.
- Approach to modeling was project driven (no larger approach or model was used)
- No real validating (with validating tool "jLive") happened – with might have been a mistake
- Project was pushed at the beginning and responsible personal had a tool-centric mind meaning the actual process-thinking was rather neglected or not focused.
- With new projects this should be avoided, better to go clean with process modeling
- Reactions on validating session were welcomed (due to the simplicity of the tool)
- Size of project or modeling teams varied, depending on the customer, between 2 and 15 people.
- Problem with larger groups all at once in one room is that – especially in S-BPM methodology – not that optimal people will get bored. It might be better to try to keep groups rather small and interconnected
- Usually there are already "project teams" with initial knowledge about the process. Others come in later – partly only for validating or details. Initial knowledge about process was given by the "project team" members – who usually knew prior about details of the process.
- With one project (BMW) review with departments took place, but made task long, because of repeatedly new inputs, contradicting other knowledge inputs.
- There was usually a kind of "process owner" in the project team of the customer who had the power to say things, or where to stop.
- Modeling team was usually only one person, with a half-modeler/half-technician joining in at later stages.
- Modeler is usually the one knowing best about the process and is needed to coordinate and give input into tool development (refinements in Metasonic)

- Pair-modeling is not so good. C. can work better alone. Can better concentrate alone.
- Review of the model directly from the customer, no often no internal reviews were necessary.
- Working alone forces modeler to work more carefully because it is his own responsibility
- No real experience at Metasonic with processes large enough to have more than one modeler.
- With larger projects with more than one modeler of course it is necessary to agree on interfaces and naming of messages in between.
- Necessary also to coordinate beforehand roughly on the detail level
- Modelers need to come together to coordinate such efforts
- The IT-System development part /technical aspects were the main reasons behind delays in most projects. Modeling itself was faster. Reasons mainly were the continued parallel development of the modeling tools and workflow engine.
- On scheduling: modeling usually is done rather quickly (1-2 days for an experienced modeler)
- Modeling should be done in rather “one scoop” and/or at points were breaks can be placed good. If not it requires some time to get “back into the model” even for the guy who created the model.
- Problematic projects: customer had set up a project team without any clue about IT or process modeling etc. – methodology was not understood by the project owner. A little basic IT know how is still required. Maybe seen a process model. Project was not a success mainly because customer wanted actually a different thing/tool.
- Positive easy at a project with AUDI was that the processes were actually already analyzed.
- It is always hard to start from scratch. It is a decisive factor that determines time of a project.
- For a project usually the customer-side needs to learn about what they actually can do. The earlier the customer realizes what is happening
- Customer understanding (of function of process modeling, and IT system development) is essential.

Interview with Florian Strecker

Function: Technical Consultant at Metasonic AG – company

General topic: experience in modeling projects (wit S-BPM)

Interview language: German

Time and place: 15th May 2010 – a restaurant in Pfaffenhofen a.d. Ilm , Germany.

Duration: 1h 7 minutes

English Summary /interpretation of the interview:

Florian Strecker has extensive work experience (3 years) in the field of S-BPM methodology, first from developing Metasonic S-BPM Suite and later with modeling business process in projects (3 larger – with a car manufacturer, and two financial/IT integration service providers) , and many smaller ones with customer interaction. All his statements were in reference to projects with Metasonic S-BPM Suite and methodology.

- Technical details of Metasonic S-BPM Suite: Subjects do have input pools. S-BPM is usually asynchronous processes (of each subject) that are linked vie messages. Limiting input pool to zero messages allows for modeling synchronous communication and processes. Not too relevant for day-to-day business though, because there Processes usually run asynchronous anyway.
- Switched from development to consulting because people with technical expertise were needed to talk to customers
- Project with car maker: Was more technical versed. Idea to build employee portal integrating many process, and integrating technology. Project team about 20-30 people, mostly technicians. Modeling itself maybe only 5 people and only “simple projects”.
- That Environment was special designed to use Process modeling to build complete applications. Process engine used not so much for data, but to handle process instances: result was a practical very good running process engine.
- Modeler still needed to know about functionality of portlets - and of course their use in the process
- process with is an example how not do, or maybe how to do:
- Initial planning was for a few months by now it is in the third year. More precise the project itself is finished and in its improvement cycle.
- The process used there is in close to its original state (before workflow management). Oriented to the original process.
- Original as it is the case in most companies: outlook and telephone, documented on power point slides. Outlook is the most used workflow tool in the world (and Excel the most used data-base tool). So everybody could do anything all the time.
- Changes from there on where often on technical side. That is also the main re
- Many phases: team building is the first.

- To Mr. Streckers Knowledge. Only Metasonic can represent such team building processes.
- Reasons for the long time the project took are mostly due to technical reasons and many requirements to the IT system. Requirements
- People today work in processes, but most do not realize that on personal level. Though he is not sure if that is even important). As a result though people to work in “applications”. They want a button to press, a UI were you can do what you need to do, very fast and very efficiently. With that process they are getting there.
- But users usually get where (they think) they want to get or try to: introducing the workflow system made people change some behavior but not in an expected way. As a result further changes where needed.
- Goal deep down (at the working level) should be to guide the users because he actually does not need to know so much.
- In any cases process management is always also knowledge management. To teach and show the people the workflow. If documented correctly.
- JPass can also be used only as a process documentation tool. But then a classical danger arises (like with ARIS): you paint a “tapestry” of the process as you think it is running and if you show it to the people they will only shake their heads.
- Also problem with “only” documentation: documentation is done, and then disappears in some closet. But Process does live, it changes, and nowadays pretty fast. And after a while the documentation does not match the documentation.
- Changes in the process do happen, speed depends on the organization. With bureaucrats it usually changes more slowly than in a small company that is growing rapidly.
- General disclaimer: you never will capture every single aspect of a process – that is why some people believe that workflow systems never really work out
- You could of course model the world. But models are not made to cover the world but to make reality easier to understand.
- How process changes were captured: usually users came to the local project manager/Product Owner/the guy responsible and told him their concerns and problems and he forwarded it to Metasonic – somewhere along the way.
- But some realizations about their own process did happen at that company and probably a few more will follow.
- Customers usually do not tell you that they have a problem, but tell you they want something to change. They change request comes usually from their idea how to solve that problem. But modeler/developer need to realize the actual cause of the problem (example from a process was given)
- Ideal case: the process owner can realize the reasons and changes by himself. Personal congruency with modeler.
- Project team size: 4 people by Metasonic, and 3-4+ at customers (test users) etc.. Around 10 persons in general.
- But process was never validated (played through) which turned out to be a problem /was a mistake (because of issues only learned later). Without Validating Workshop Florian would not like to do further projects. More than one workshop may be advisable but in real life
- Process validating is one part, but (technical) service integration (at a later stage) is another. That later part is harder and not easy to do. But at that time you can also check again for process structures.
- Initial projects (including setting up infrastructure etc.) take (of course) longer than and require more resources than later modeling projects.

- Modeling with Metasonic is possible in more than one direction: of course official way: find subjects, find messages (communication), and then define individual work flow. Alternative way: find at least one person involved in the process, get their workflow, derive partners from that, and afterwards get their workflow, communication between those two is a quasi automatic result, but there is quite a big chance that the two workflows (communication via messages) will not match 100% because of different point of views.
- On Separation of a process into subjects: “cuts” (between internal and external should be dependent on where I need to see further communications or not
- Modeler needs to keep overview (the user does not necessarily)
- If one process involves communication with more than 9 different subjects (not including multi subjects) maybe you should think over your process. In such a case it could be suspected that it is actually more than one process (with different participants).
- About project and resource management: coordination does eat up your time, because of communication overhead. Applicable also for business processes as well.
- In Aris there is only on process flow, making parallel work in Aris hard to get model parallel work.
- War story from the trenches: company were there was rule that no EPK was supposed to be larger than one Din A4 sheet of paper, consecutively every EPK was indeed on one sheet of A4 paper, but with links to other parts of the process at start and end. Moral of the story: separation of models for Formal reasons is not necessarily useful.
- There is no general rule for how to separate a model. Depends on a good modeler. And there is not the one solution.
- Good point for separation is functional areas in a process.
- Most problems were due to technical problems
- Problematic is if the focus of project members is different. If not in process mind
- Problem in big companies. There are Aris-using organizers/analysts (people) who model with an Aris-frame of mind. And there are the departments. Both usually have not too much to do with each other – which is a problem by itself. Those guys though do not like the rather low-level modeling, because they want to see their hierarchies, to set KPIs or something alike.
- About S-BPM and process hierarchies: Communication is no hierarchy by itself.
- S-BPM concept can go far beyond what current existing tool is capable of.

Function: works in project management for IT department of a large German financial institution in Frankfurt a.M.

General topic: process modeling projects

Interview language: German

Time and place: 13th July 2010 – University of Karlsruhe

Duration: 42 minutes

English Summary /interpretation of the interview:

- Has studied/training for three years and has five year working experience
- Working experience:
- Process Modeling in the trade sector in larger and smaller projects
- Process and requirements recording for a substantial process covering several departments
- ARIS tool standard were used.
- Process Models used/required:
- Organigram ,A view on information-carriers as well as application systems and the in and output
- These Models grew while recording the process, continuous growing, step by step and same or similar for all projects
- Process models grow gigantic, many involved departments and business areas – can get very complex

→ It is impossible to capture all aspects once and previous

- people usually do not even know how complex their own work is and realize this only while talking about it
- people do not work really in "processes" , they do not have the mind for that.
- Modeling dependencies is the goal, not just "as people do it" → need to always ask: why do you do it that way and could it be done differently – parallel, in different order etc.?
- It was new for the people to get into this kind of thinking.
- Many were against the process modeling projects, it not necessary
- Crass differences in expectancies for time: expected to be done in five minutes and it took days.
- This needs to be understood
- There has been a formal decision to record and model projects, as such the departments were required (forced) to supply the (time) resources.
- Conflict between daily operations and recording/interview task – an initial source of hostile attitudes towards such a project
- Later realization: the analyzing and talking and thinking through about their processes also was beneficial for the departments themselves: as documentation

for new employees, as basis for other requirements for e.g. new software. Better than starting such from scratch

- Beneficial thoughts (for people in the process that they were required to think for the first time): what is happening before and after me? Extreme example: people realized they were doing the same thing in two different departments, realizing certain things were not necessary
- This is simply reality. This is not just bad organization
- Motivation for modeling projects: Not thinking about own processes before, about what one is actually doing let to giving not all or complete requirements for other software projects, which let to dissatisfaction with the software tools they were used for
- Reasons: misunderstandings about wording (different wordings for same parameters e.g.), no understanding over the whole horizon or "bandwidth" of task involved.
- What took most time: separation/cuts – how to separate the process to satisfy all participants, and do it meaningfully – it took weeks.
- Once you have an (ARIS) model, it is hard to change it all afterwards. Need to work neatly from the beginning
- It also takes time in the beginning, to decide on a detail level. Need to be thought about how deep down you model, (depending in the goal of modeling project)
- Project managing: had project manager and coordinators, people doing only the modeling, also moderators (usually the same guys as the modelers), quality assurance (technical as well as content dependant) and of course departments for input
- Big group interviews with the departments to come up with definitions (roughly 15 people)
- Afterwards modeling in smaller groups (2-4), tried to keep modeling groups small
- Beamer, modeler/moderator, one room, asking questions and directly within the modeling tool (required the modeler to be very knowledgeable about the whole process)
- Modeler need to be coordinated, if more than one person they need to know who put what, when and why into the model
- Responsibilities were widely distributed (modeling, quality assurance, acceptance)
- Quality assurance: first step: modeler for checked for formal and standard compliance to standards, afterwards technician checked for technical correctness, external checks by ARIS company, and of course departments checked for correctness of model content. Departments at the end
- At the beginning departments needed to be helped to understand things in the models (how is this meant, what is written there in the model – especially involving connectors) later, when they were experienced it was easier.
- Timeframe for process project covering many departments took years. For smaller projects it might be faster though.
- For process recording with the departments, IT department actually went to the involved persons. Nowadays online/tele-working is possible, but face-to-face is the best to explain modeling procedures, to overcome reluctant and hostile attitudes towards a project
- Web-based working is of course possible, but it requires "trained" and open-minded persons, and only for task with few people involved (not for bigger processes in the beginning)

- People from same department have to discuss about what is “right”. Modeling project does need to give them a forum for that – best on-site being a moderator. Even more important if you two different departments are involved.
- Especially in the beginning, moderating is very important, when two different departments clash. Try show possible solution, under a complex, non-complete information situation
- There is often human interaction where you have to get one common view on it. Modeler need to find the point where it is neutral
- No tool will take away the need for persons that have a total overview about a process. Something no single person can achieve that.
- On S-BPM and Metasonic: tool is good, but buggy, the idea – as perceived by Stephan – though
- Separating into “Subjects” maybe sometime very good, sometime not
- Separation of process is always needed, not matter what modeling paradigm,
- You have to think about it practically before you start about: what do I want, how do I split up/separate the processes, how to I model (especially if I want to change something)
- ARIS is not really suitable if you want to directly turn your model into a work-flow supporting tool. Metasonic S-BPM suite strength is to directly use the model for that purpose.

Bibliography

[ArGDS07] Arisholm, Erik, et al. 2007. Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise. *IEEE Transactions on Software Engineering 33 (2).* February 2007, pp. 65–86.

[Beck06] Beck, Kent. 2006. *Extreme programming explained: embrace change.* Boston : Addison-Wesley, 2006. ISBN: 978-0-321-27865-4.

[Beck01] —. 2001. Manifesto for Agile Software Development. *agilemanifesto.org/.* [Online] 2001. [Cited: August 7, 2010.] http://agilemanifesto.org/.

[Bogn09] Bogner, Alexander. 2009. *Experteninterviews : Theorien, Methoden, Anwendungsfelder.* Wiesbaden : Verlag für Sozialwissenschaften, 2009. ISBN: 978-3-531-16259-1.

[Buch09] Buchwald, Hagen. 2009. *The Power of as-is processes - Speech records.* S-BPM One Conference in Karlsruhe 22nd October 2009 : s.n., 2009.

[Cijo08] Cijosouw, R.S. and al., et. 2008. Omissions in Managing Knowledge in Innovation Processes or How to Handle knowledge, Humans and Taks: A Semio-Cognitive Approach. [book auth.] Pierre-Jean [publisher] Charrel. *Project Management and Risk Managemet in Complex Projects - Studies in Organizational Semiotics.* Dordrecht : Springer, 2008, pp. 15-45.

[Cohn10] Cohn, Mike. 2010. *Succeeding with Agile : software development using Scrum.* Upper Saddle River, NJ : Addison-Wesley, 2010. ISBN: 978-0-321-57936-2.

[CoSiSu09] Coman, Irina Diana, Sillitti, Alberto and Succi, Giancarlo. 2009. Investigating the Usefulness of Pair-Programming in a Mature Agile Team. [book auth.] Wil / van der Aalst, et al. *Agile Processes in Software Engineering and Extreme Programming.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2009, pp. 128-136.

[Demi00] Deming, William Edwards. 2000. *Out of the crisis.* Cambridge, Mass. [u.a.] : MIT Press, 2000. ISBN: 0-262-54115-7.

[Dörn09] Dörner, Dietrich. 2009. *Die Logik des Misslingens : strategisches Denken in komplexen Situationen.* Reinbek bei Hamburg : Rowohlt, 2009. ISBN: 978-3-499-61578-8.

[Fric10] Frick, Marike. 2010. Interview mit Roland Springer: Mitarbeiter an die kurze Leine! *manager-magazin.de.* [Online] 28. 7 2010. [Zitat vom: 4. 8 2010.] http://www.manager-magazin.de/unternehmen/artikel/0,2828,708477,00.html.

[Funk92] Funke, Joachim. 1992. *Wissen über dynamsiche Systeme: Erwerb, Repräsentation und Anwendung.* Heidelberg : Springer, 1992. ISBN-10: 354-055-2235.

[GaLiHa07] Gan, M., Liu, K. and Han, B. 2007. Semiotic Transformations from Business Domain to IT domain in Information Systems Development. [book auth.] Pierre-Jean [publisher.] Charrel. *Project management and risk management in complex projects : studies in organizational semiotics.* Dordrecht : Springer, 2007, pp. 133-143.

[Gran09] Grandin, Greg. 2009. *Fordlandia.* New York : Metropolitan Books, 2009.

[HaDAS09] Hannay, Jo E., et al. 2009. The effectiveness of pair programming: A meta-analysis. *Information and Software Technology 51 (7).* July 2009, pp. 1110-1122.

[Hess82] Hesse, Friedrich W. 1982. Effekte des semantischen Kontexts auf die Bearbeitung komplexer Probleme. *Zeitschrift für experimentelle und angewandte Psychologie 29.* 1 1982, S. 62-91.

[Hürl09] Hürlimann, Mark. 2009. *Dealing with Real-World Complexity : Limits, Enhancements and New Approaches for Policy Makers.* Wiesbaden : Gabler Verlag / GWV Fachverlage GmbH, Wiesbaden, 2009. ISBN: 978-3-8349-8074-8.

[ISAC10] ISACA. 2010. Official website of the Information Systems Audit and Control Association. *Official website of the Information Systems Audit and Control Association.* [Online] 2010. [Cited: September 21, 2010.] http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx.

[ITIL10] ITIL® Official Website. 2010. Official ITIL® Website. *Official ITIL® Website.* [Online] 2010. [Cited: August 21, 2010.] http://www.itil-officialsite.com/home/home.asp.

[JoEwUh08] John, M., Ewald, R. and Uhrmacher, A. M. 2008. A Spatial Extension to the Pi Calculus. *Electronic Notes in theoretical Computer Science 194.* 2008, pp. 133-148.

[Jose94] Josephson, John R. 1994. *Abductive inference : computation, philosophy, technology.* Cambridge : Cambridge Univ. Press, 1994. ISBN: 0-521-43461-0.

[Knib09b] Kniberg, Henrik and Skarin, Mattias. 2009. Kanban and Scrum - making the most of both. *Info Q.* [Online] 12 21, 2009. [Cited: 8 6, 2010.] http://www.infoq.com/minibooks/kanban-scrum-minibook.

[Knib09] Kniberg, Henrik. 2009. Kanban vs Scrum. *www.crisp.se.* [Online] 06 29, 2009. [Cited: 8 7, 2010.] http://www.crisp.se/henrik.kniberg/Kanban-vs-Scrum.pdf.

[KvBr09] Kvale, Steinar and Brinkmann, Svend. 2009. *InterViews : learning the craft of qualitative research interviewing.* Los Angeles [a.o.] : Sage, 2009. ISBN: 978-0-7619-2541-5.

[Larm10] Larman, Craig and Vodde, Bas. 2010. *Practices for scaling lean & agile development : large, multisite, and offshore product development with large-scale Scrum.* Upper Saddle River, NJ : Addison-Wesley, 2010. ISBN-13: 978-0321636409.

[Larm09] Larman, Craig. 2009. *Scaling lean & agile development: thinking and organizational tools for large- scale scrum.* Upper Saddle River,NJ : Addison-Wesley, 2009. ISBN: 978-0-321-48096-5.

[Lien99] Liening, Andreas. 1999. *Komplexe Systeme zwischen Ordnung und Chaos : neuere Entwicklungen in der Theorie nicht-linearer dynamischer Systeme und die Bedeutung für die Wirtschaftswissenschaft und ihre Didaktik.* Münster : Lit, 1999. ISBN: 3-8258-4023-9.

[Like04] Liker, Jeffrey K. 2004. *The Toyota way : 14 management principles from the worlds greatest manufacturer.* New York [a.o.] : McGraw-Hill, 2004. ISBN: 0-07-139231-9.

[Likn07] Likner, Jeffrey. 2007. *Der Toyota Weg: 14 Mangementprinzipien des weltweit erfolgreichsten Automobilekonzerns (German translation).* [Übers.] Almuth Braun. München : FinanzBuch-Verlag, 2007. ISBN: 978-3-89879-188-5.

[LiST07] Liu, K., Sun, L. and Tan, S. 2007. Using Problem Articulaton Method to Assist Planning and Management of Complex Projects. [book auth.] Pierre-Jean [publisher.] Charrel. *Project management and risk management in complex projects : studies in organizational semiotics.* Dordrecht : Springer, 2007, pp. 3-13.

[LuChNo08] Lui, Kim Man, Chan, Keith C. C. and Nosek, John T. 2008. The Effect of Pairs in Program Design Tasks. *IEEE Transactions on Software Engineering 32.* March/April 2008, pp. 197-211.

[Meta10] Metasonic AG. 2010. Metasonic AG Company Website. *Metasonic AG Company Website.* [Online] August 2, 2010. [Cited: August 2, 2010.] http://www.metasonic.de.

[Mill56] Miller, George. 1956. The Magical Number Seven, Plus or Minus Two - Some Limits on Our Capacity for Processing Information. *Psychological Review Vol. 63.* 1956, pp. 343-355.

[Mil04] Milner, Robin. 2004. *Communicating and mobile systems: the pi-calculus.* Cambridge : Cambridge Univ. Press, 2004. ISBN: 0-521-64320-1.

[Müll10] Müller-Michaels, Harro. 2010. Schlechtes Image? Letzte Chance. *Die Zeit.* [Online] 5 12, 2010. [Cited: 9 19, 2010.] http://www.zeit.de/2010/20/C-Mueller-Michaels.

[Nou98] Nourse, Lee. 1998. Japanese and English Speech Codes in Cross-Cultural Communications. *EastAsianReview 2.* 1998, pp. 81-92.

[Oest06] Oestereich, Bernd. 2006. Timeboxing - das Rückrat agiler Projecte. *Agiles Projektmanagement : Beiträge zur Konferenz interPM Glashütten 2006.* Heidelberg : dpunkt-Verl., 2006, S. 53-68.

[PlSt97] Platon and Stokes, Micheall. 1997. *Apology.* Warminster : Bertrams, 1997. ISBN: 0-85668-371-X.

[Proc10] Process Lab. 2010. Process Lab Page. *Franfurt School of finance.* [Online] August 15, 2010. [Cited: August 15, 2010.] http://www.frankfurt-school.de/content/en/processlab.

[Puhl06] Puhlmann, Frank. 2006. Why do we actually need the Pi-Calculus for Business Process Management? *Website of the Hasso Plattner Institute of University of Potsdam.* [Online] 2006. [Cited: September 21, 2010.] http://bpt.hpi.uni-potsdam.de/pub/Public/FrankPuhlmann/BIS2006-PIC.pdf.

[Reic03] Reichertz, Jo. 2003. *Die Abduktion in der qualitativen Sozialforschung.* Opladen : Leske + Budrich, 2003. ISBN: 3-8100-3595-5.

[RoSh10] Robinson, Hugh and Sharp, Helen. 2010. Collaboration, Communication and Coordination in Agile Software Developement Practice. [book auth.] Ivan Mistrík. *Collaborative Software Engineering.* Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2010, pp. 93-108.

[ScFlGi10] Schmidt, Werner, Fleischmann, Albert and Gilbert, Oliver. 2010. Subject-Oriented Business Process Management. *www.jCom1 AG Companay Website.* [Online] August 2, 2010. [Cited: August 2, 2010.] http://www.jcom1.com/cms/fileadmin/media/news/Artikel/Beitrag_englisch_v4.pdf.

[ScFlGi09] —. 2009. Subject-Oriented Business Proces Management. *HMD - Praxis der wirtschafsinformatik.* April 2009, 266, S. 52-62.

[Sch08] Schulz von Thun, Friedemann. 2008. *Miteinander reden - 1: Störungen und Klärungen : allgemeine Psychologie der Kommunikation.* Reinbek : Rowohlt Taschenbuch, 2008.

[Sch03] —. 2003. *Praxisberatung in Gruppen : erlebnisaktivierende Methoden mit 20 Fallbeispielen.* Weinheim : Beltz, 2003.

[Sch04] Schulz von Thun, Friedemann, Ruppel, Johannes and Stratmann, Roswitha. 2004. *Miteinander reden: Kommunikationspsychologie für Führungskräfte.* Reinbek bei Hamburg : Rowohlt, 2004.

[Schw04] Schwaber, Ken. 2004. *Agile project management with Scrum.* Redmond, Wash. : Microsoft Press, 2004. ISBN-13: 978-0-7356-19937.

[ScSu10] Schwaber, Ken and Sutherland, Jeff. 2010. Scrum.org. *http://www.scrum.org.* [Online] February 1, 2010. [Cited: February 6, 2010.] http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf.

[Scot02] Scott, Kendall. 2002. *The unified process explained.* Boston, Mass. : Addison-Wesley, 2002. ISBN: 0-201-74204-7.

[Silv09a] Silver, Bruce. 2009. *BPMN - method and style.* Aptos, Calif : Cody-Cassidy Pr., 2009. ISBN: 978-0-9823681-0-7.

[Toy06] Toyota Tradtions. 2006. Toyota Traditions. *Toyot international.* [Online] March/April 2006. [Cited: 8 15, 2010.] http://www2.toyota.co.jp/en/vision/traditions/mar_apr_06.html.

[AlHKB03] van der Allst, W.M.P, et al. 2003. Workflow Patterns. *Distributed and Parallel Databases,.* July 2003, 14 (3), pp. 5-51.

[Veri08] VersionOne. 2008. 3rd Annual Survey: 2008 - “The State of Agile Development”. *VersionOne Company Homepage.* [Online] 2008. [Cited: August 23, 2010.] http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.

[Vest02] Vester, Frederic. 2002. *Die Kunst vernetzt zu denken - Ideen und Werkzeuge für einen neuen Umgang mit Komplexität.* Münschen : Deutscher Taschenbuch Verlag, 2002. ISBN: 978-3-423-33077-0.

[WiFi10] Wikipedia - Five Ws. 2010. Five Ws. *Five Ws.* [Online] August 4, 2010. [Cited: August 14, 2010.] http://en.wikipedia.org/wiki/Five_Ws.

[Wik10] Wikipedia - Scrum. 2010. Scrum. *Wikipedia.org.* [Online] August 7, 2010. [Cited: August 7, 2010.] http://en.wikipedia.org/wiki/Scrum_%28development%29.

Table of Figures